UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF WISCONSIN

Hewlett-Packard Development Company, L.P.,

Plaintiff,

Case No. 04-C-0789-C

v.

eMachines, Inc.,

Defendant.

SECOND AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff, Hewlett-Packard Development Company, L.P., for an amended complaint against defendant, eMachines, Inc., alleges as follows:

JURISDICTION

1. This is an action for patent infringement. This Court has jurisdiction under 28 U.S.C. §§ 1331, 1338, and pursuant to the patent laws of the United States of America, 35 U.S.C. § 100, *et seq.*

VENUE

2. Venue properly lies within the Western District of Wisconsin pursuant to the provisions of 28 U.S.C. §§ 1391 (b), (c) and 1400(b).

THE PARTIES

3. Hewlett-Packard Development Company, L.P. ("HPDC") is a limited partnership formed under the laws of Texas and has its principal place of business at 20555 State Highway 249, Houston, Texas 77070.

4. eMachines, Inc. ("eMachines") on information and belief, is a Delaware corporation and has its principal place of business at 14350 Myford Road, Suite 100, Irvine, California 92606.

5. Upon information and belief, eMachines transacts business in this judicial district, including the sale and offering for sale of its products, and eMachines has sufficient contacts with this judicial district to subject eMachines to the jurisdiction of this Court.

FACTUAL BACKGROUND

6. Hewlett-Packard ("HP") is an industry leader in innovation in the personal computer and server product development areas. HP, including its acquired companies, has made major investments in its intellectual property. HP protects its inventions and requires fair and appropriate value when other companies use its intellectual property.

7. HPDC is a limited partnership and wholly-owned subsidiary of HP. HPDC is a holding company for HP intellectual property rights and is the intellectual property licensing entity for HP worldwide.

8. On October 24, 2000, United States Patent No. 6,138,184 ("the '184 Patent"), titled *System for Parallel Port with Direct Memory Access Controller for Developing Signal to Indicate Packet Available and Receiving Signal That Packet Has Been Accepted*, was duly and legally issued. A true and correct copy of the '184 Patent is attached hereto as Exhibit A.

9. HPDC is the assignee and the owner of all right, title and interest in and to the '184 Patent. Accordingly, HPDC has the right to bring this suit for damages and injunctive relief.

10. On May 15, 2001, United States Patent No. 6,233,691 ("the '691 Patent"), titled *Apparatus for Reducing Computer System Power Consumption*, was duly and legally issued. A true and correct copy of the '691 Patent is attached hereto as Exhibit B.

11. HPDC is the assignee and the owner of all right, title and interest in and to the '691 Patent. Accordingly, HPDC has the right to bring this suit for damages and injunctive relief.

12. On August 20, 2002, United States Patent No. 6,438,697 ("the '697 Patent"), titled *Demand-Based Processor Clock Frequency Switching*, was duly and legally issued. A true and correct copy of the '697 Patent is attached hereto as Exhibit C.

13. HPDC is the assignee and the owner of all right, title and interest in and to the '697 Patent. Accordingly, HPDC has the right to bring this suit for damages and injunctive relief.

COUNT I
INFRINGEMENT OF '184 PATENT

14. HPDC incorporates by reference the allegations set forth in paragraphs 1 through 13.

15. HPDC is informed and believes, and on that basis alleges, that eMachines is infringing the '184 Patent by making, using, offering for sale, and/or selling within the United States devices that embody the inventions disclosed and claimed in the '184 Patent, and/or by importing into the United States devices that embody the inventions disclosed and claimed in said patent.

16. In addition to direct infringement, HPDC is informed and believes, and on that basis alleges, that eMachines has induced and contributed to infringement by others of the '184 Patent.

17. Based upon information and belief, eMachines has notice that it is manufacturing and selling products that infringe the '184 Patent. Despite such notice, eMachines has continued to willfully infringe said patent by making, using, offering to sell, and/or selling within the United States products that embody the inventions disclosed and claimed in said patent, and/or by importing such products into the United States.

18. HPDC has been irreparably harmed by eMachines' acts of infringement, and will continue to be harmed unless and until eMachines' acts of infringement are enjoined and restrained by order of this Court. HPDC has no adequate remedy of law.

19. As a result of eMachines' acts of infringement, HPDC has suffered and will continue to suffer damages in an amount to be proven at trial.

20. This case is an "exceptional" case within the meaning of 35 U.S.C. § 285.

COUNT II
INFRINGEMENT OF '691 PATENT

21. HPDC incorporates by reference the allegations set forth in paragraphs 1 through 20.

22. HPDC is informed and believes, and on that basis alleges, that eMachines is infringing the '691 Patent by making, using, offering for sale, and/or selling within the United States devices that embody the inventions disclosed and claimed in the '691 Patent, and/or by importing into the United States devices that embody the inventions disclosed and claimed in said patent.

23. In addition to direct infringement, HPDC is informed and believes, and on that basis alleges, that eMachines has induced and contributed to infringement by others of the '691 Patent.

24. Based upon information and belief, eMachines has notice that it is manufacturing and selling products that infringe the '691 Patent. Despite such notice, eMachines has continued to willfully infringe said patent by making, using, offering to sell, and/or selling within the United States products that embody the inventions disclosed and claimed in said patent, and/or by importing such products into the United States.

25. HPDC has been irreparably harmed by eMachines' acts of infringement, and will continue to be harmed unless and until eMachines' acts of infringement are enjoined and restrained by order of this Court. HPDC has no adequate remedy of law.

26. As a result of eMachines' acts of infringement, HPDC has suffered and will continue to suffer damages in an amount to be proven at trial.

27. This case is an "exceptional" case within the meaning of 35 U.S.C. § 285.

COUNT III
INFRINGEMENT OF '697 PATENT

28. HPDC incorporates by reference the allegations set forth in paragraphs 1 through 27.

29. HPDC is informed and believes, and on that basis alleges, that eMachines is infringing the '697 Patent by making, using, offering for sale, and/or selling within the United States devices that embody the inventions disclosed and claimed in the '697 Patent, and/or by importing into the United States devices that embody the inventions disclosed and claimed in said patent.

30. In addition to direct infringement, HPDC is informed and believes, and on that basis alleges, that eMachines has induced and contributed to infringement by others of the '697 Patent.

31. Based upon information and belief, eMachines has notice that it is manufacturing and selling products that infringe the '697 Patent. Despite such notice, eMachines has continued to willfully infringe said patent by making, using, offering to sell, and/or selling within the United States products that embody the inventions disclosed and claimed in said patent, and/or by importing such products into the United States.

32. HPDC has been irreparably harmed by eMachines' acts of infringement, and will continue to be harmed unless and until eMachines' acts of infringement are enjoined and restrained by order of this Court. HPDC has no adequate remedy of law.

33. As a result of eMachines' acts of infringement, HPDC has suffered and will continue to suffer damages in an amount to be proven at trial.

34. This case is an "exceptional" case within the meaning of 35 U.S.C. § 285.

WHEREFORE HPDC requests a judgment be entered against eMachines as follows:

A. That HPDC be adjudged the owner of the '184 Patent, the '691 Patent, and the '697 Patent and entitled to all rights of recovery thereunder, and that such patents are valid and enforceable;

B. That eMachines be adjudged to have infringed, induced infringement and contributed to infringement of the '184 Patent, the '691 Patent, and the '697 Patent;

C. That eMachines, its officers, principals, agents, attorneys, servants, employees and all others acting by or under their direction and authority, and their successors and assigns, be enjoined by preliminary and permanent injunctions from making, using, offering to sell or selling in the United States any infringing products or any other product substantially equivalent thereto which is also within the scope of any claim of the '184 Patent, the '691 Patent,

and the '697 Patent and from importing into the United States any infringing products or any other product substantially equivalent thereto which is also within the scope of any claim of the '184 Patent, the '691 Patent, and the '697 Patent;

D. That HPDC be awarded an accounting for and recovery of damages under 35 U.S.C. § 284 adequate to fully compensate it for infringement by eMachines of the '184 Patent, the '691 Patent, and the '697 Patent in an amount to be proven at trial;

E. That HPDC be awarded treble damages in view of the reckless, willful and deliberate nature of eMachines' infringement, pursuant to 35 U.S.C. § 284;

F. For costs and attorneys' fees pursuant to 35 U.S.C. § 285;

G. For interest thereon at the legal rate; and

H. For such other and further relief as the Court may deem just and proper.

JURY DEMAND

Plaintiff demands a trial by jury on all issues properly tried to a jury.

Dated: December 17, 2004

Allen A. Arntsen (SBN 1015038)
Naikang Tsao (SBN 1036747)
Mark L. Langenfeld (SBN 1009394)
FOLEY & LARDNER LLP
150 East Gilman Street
P.O. Box 1497
Madison, WI 53701-1497
(608) 257-5035
(608) 258-4258 Fax

John Allcock
Andrew P. Valentine
M. Elizabeth Day
Barry K. Shelton
GRAY CARY WARE & FREIDENRICH LLP
2000 University Avenue
East Palo Alto, CA 94303-2248
(650) 833-2000
(650) 833-2001 Fax

Attorneys for Plaintiff
Hewlett-Packard Development Company, L.P.

EXHIBIT A

US006138184A

United States Patent [19]

Jirgal

[11] Patent Number: 6,138,184

[45] Date of Patent: *Oct. 24, 2000

[54] **SYSTEM FOR PARALLEL PORT WITH DIRECT MEMORY ACCESS CONTROLLER FOR DEVELOPING SIGNAL TO INDICATE PACKET AVAILABLE AND RECEIVING SIGNAL THAT PACKET HAS BEEN ACCEPTED**

[75] Inventor: **James J. Jirgal**, Houston, Tex.

[73] Assignee: **Compaq Computer Corporation**, Houston, Tex.

[*] Notice: This patent is subject to a terminal disclaimer.

[21] Appl. No.: **09/286,806**

[22] Filed: **Apr. 6, 1999**

Related U.S. Application Data

[63] Continuation of application No. 08/640,223, Apr. 30, 1996, Pat. No. 5,892,976, which is a continuation of application No. 08/403,585, Mar. 14, 1995, Pat. No. 5,539,917, which is a continuation of application No. 07/431,657, Nov. 3, 1989, abandoned.

[51] **Int. Cl.**[7] **G06F 13/14**

[52] **U.S. Cl.** **710/25**; 710/59; 710/64; 710/23; 709/212; 713/500

[58] **Field of Search** 710/25, 23, 59, 710/64; 709/212; 713/500

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,075,691	2/1978	Davis et al.	710/64
4,395,756	7/1983	Daniels	710/59
4,463,421	7/1984	Laws .	
4,639,910	1/1987	Toegel et al.	370/366
4,730,308	3/1988	Friedman et al.	370/276
4,755,937	7/1988	Glier .	
4,802,120	1/1989	McCoy	713/500
4,922,416	5/1990	Krishnan et al.	709/212
4,989,113	1/1991	Hull .	
5,014,237	5/1991	Masters .	
5,038,282	8/1991	Gilbert .	
5,090,830	2/1992	Kroeger et al.	400/719
5,113,523	5/1992	Colley .	
5,255,238	10/1993	Ichige et al.	365/220
5,539,917	7/1996	Jirgal	710/22
5,892,976	4/1999	Jirgal	710/22

FOREIGN PATENT DOCUMENTS

62-208121	of 1987	Japan .
62-237556	of 1987	Japan .

OTHER PUBLICATIONS

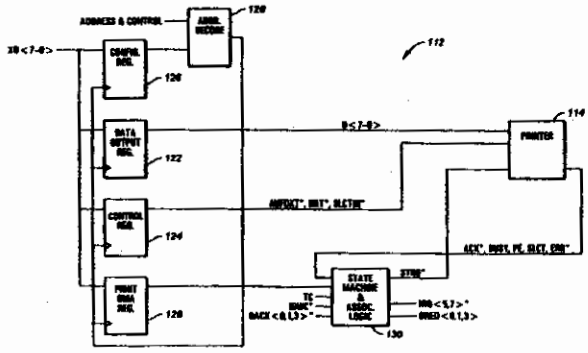
Derwent Publications Ltd., London, GB, Abstract AU8822202, Week 8924.
PC Magazine, "IBM PS/2 Models 90 and 95 True 486 Power at Last", Feb. 26, 1991, pp. 287–308.
Multi–I/O Card Users Manual, p. 23.
M. Morris Mano, Computer Systems Architecture, 1982, p. 428–434.
Motorola Semiconductor Technical Data Hemos Singie–Chip MicroComputer 1985, pp. 4–1 to 4–6.
Intel Product Guide, 1983 p. 29.

Primary Examiner—Thomas C. Lee
Assistant Examiner—Abdelmoniem Elamin
Attorney, Agent, or Firm—Akin, Gump, Strauss, Hauer & Feld, LLP

[57] **ABSTRACT**

The parallel or printer port in a personal computer can receive data from the memory under the control of the direct memory access (DMA) controller, releasing processor resources. The processor enables the parallel port, which then indicates to the DMA controller the desire to transfer data. A state machine in the parallel port, along with associated circuitry, responds to the transfer of the data to the parallel port and then controls the transfer of data to the attached device, usually a printer. The state machine causes an interrupt to the processor when the transfer is complete or on receipt of errors from the external device. The state machine also communicates with the DMA controller to repeat the transfer process until the transfer is complete or an error occurs. Various DMA channels and parallel port locations can be used. Direct transfers by the processor are blocked during DMA controller handled transfers.

51 Claims, 10 Drawing Sheets



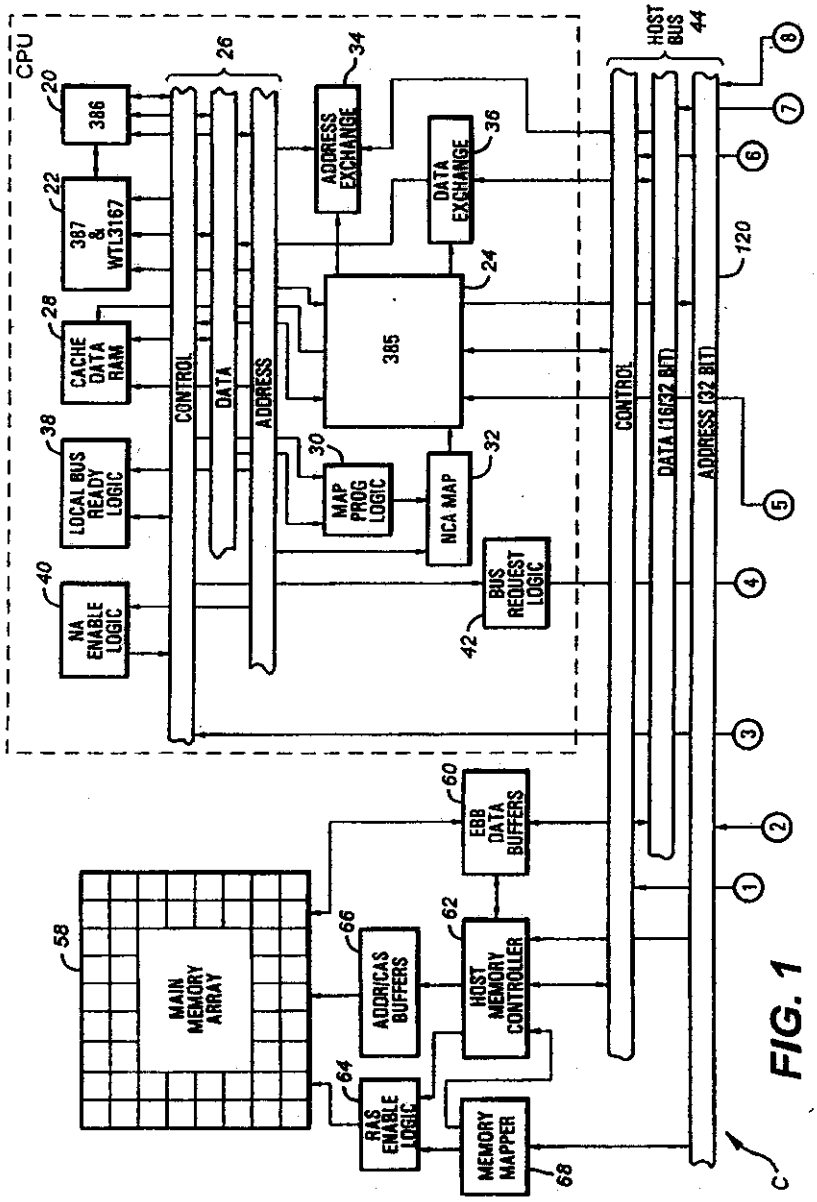
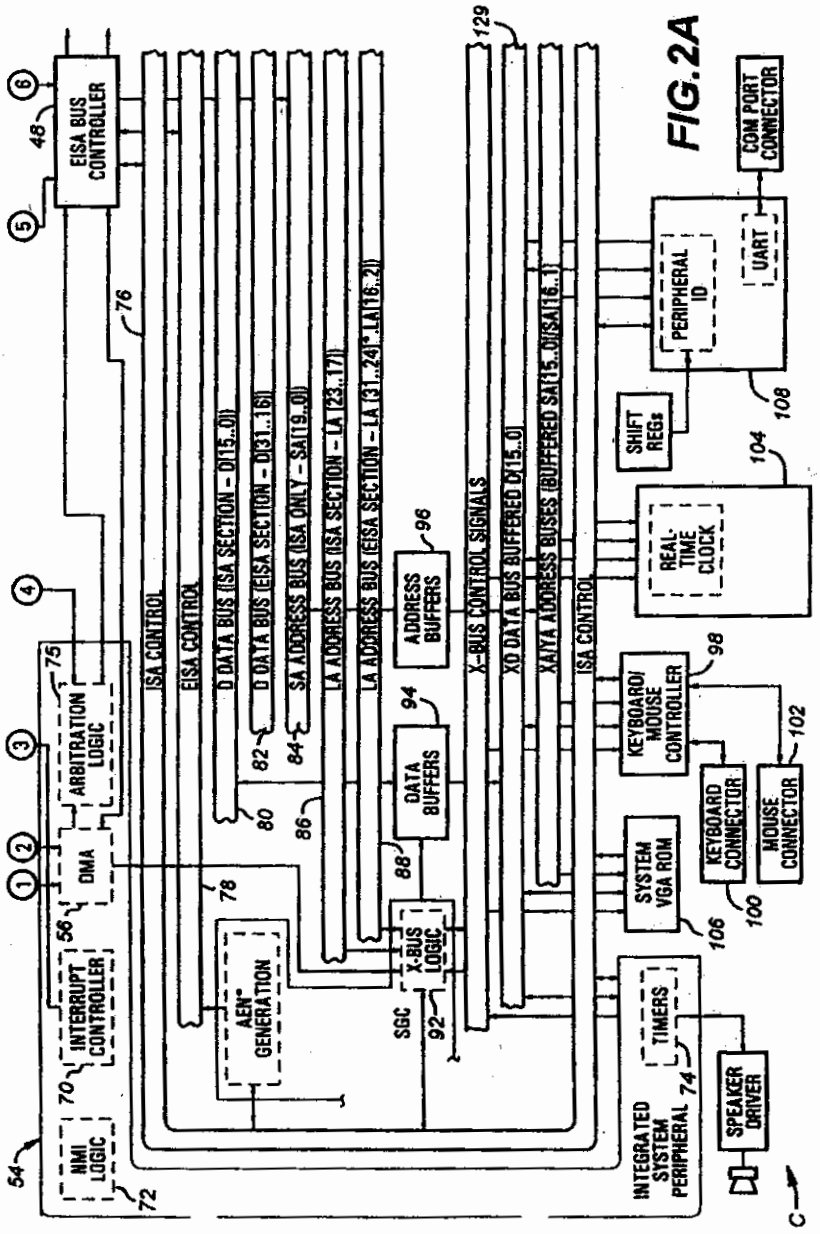

CPU
386
387 & WTL3167
CACHE DATA RAM
LOCAL BUS READY LOGIC
NA ENABLE LOGIC
CONTROL
DATA
ADDRESS
ADDRESS EXCHANGE
DATA EXCHANGE
385
MAP PROG LOGIC
NCA MAP
BUS REQUEST LOGIC
HOST BUS
CONTROL
DATA (16/32 BIT)
ADDRESS (32 BIT)
EBB DATA BUFFERS
MAIN MEMORY ARRAY
ADDR/CAS BUFFERS
HOST MEMORY CONTROLLER
RAS ENABLE LOGIC
MEMORY MAPPER

FIG. 1


EISA BUS CONTROLLER
48
ISA CONTROL
EISA CONTROL
D DATA BUS (ISA SECTION - D(15..0))
D DATA BUS (EISA SECTION - D(31..16))
SA ADDRESS BUS (ISA ONLY - SA(19..0))
LA ADDRESS BUS (ISA SECTION - LA (23..17))
LA ADDRESS BUS (EISA SECTION - LA (31..24)*.LA(16..2))
129
X-BUS CONTROL SIGNALS
XD DATA BUS BUFFERED D(15..0)
XA/YA ADDRESS BUSES (BUFFERED SA(15..0)/SA(16..1))
ISA CONTROL
ADDRESS BUFFERS
96
DATA BUFFERS
94
ARBITRATION LOGIC
75
DMA
INTERRUPT CONTROLLER
56
NMI LOGIC
70
AEN* GENERATION
78
X-BUS LOGIC
SGC
92
INTEGRATED SYSTEM PERIPHERAL
TIMERS
74
SPEAKER DRIVER
KEYBOARD/ MOUSE CONTROLLER
98
SYSTEM VGA ROM
106
KEYBOARD CONNECTOR
100
MOUSE CONNECTOR
102
SHIFT REGs
REAL-TIME CLOCK
104
PERIPHERAL ID
UART
108
COM PORT CONNECTOR
54
72
80
82
84
86
88
76
FIG. 2A

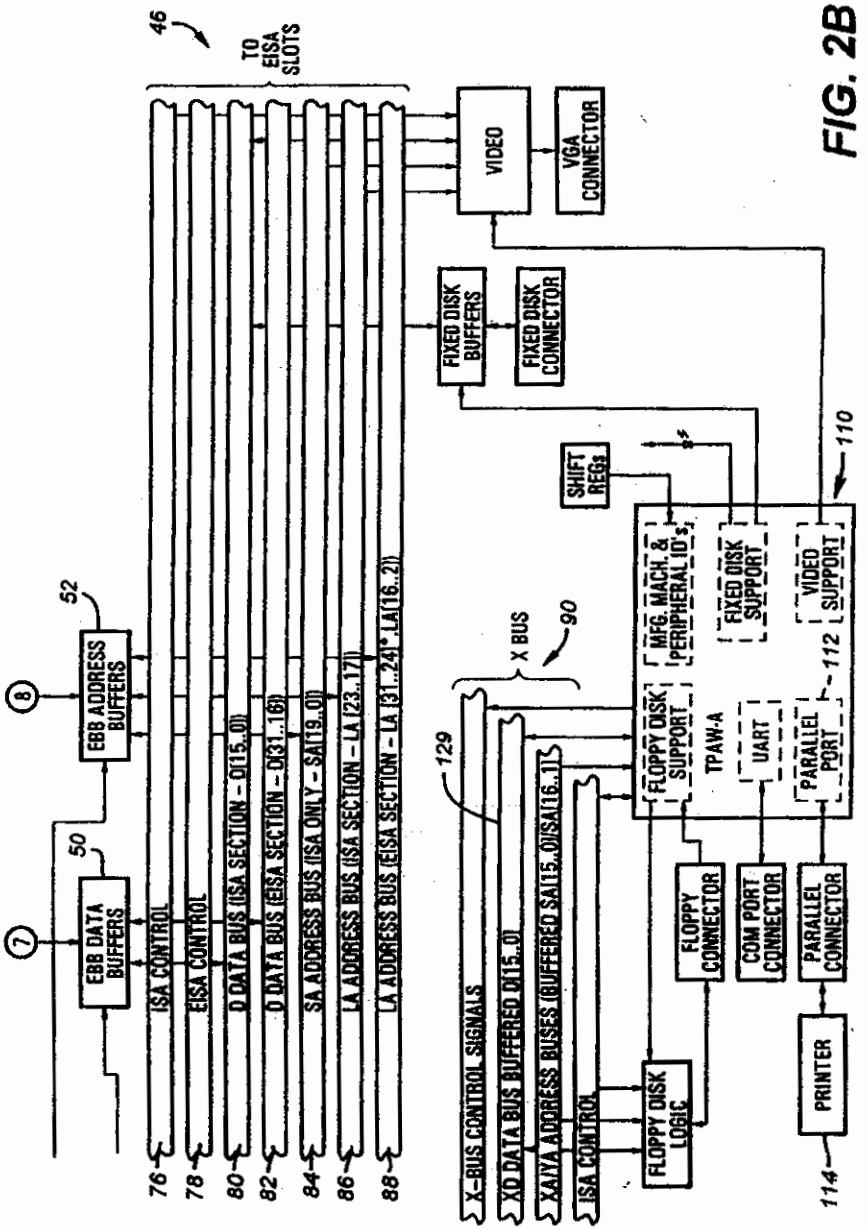

46
TO EISA SLOTS
VIDEO
VGA CONNECTOR
FIXED DISK BUFFERS
FIXED DISK CONNECTOR
SHIFT REGs
110
52
EBB ADDRESS BUFFERS
50
EBB DATA BUFFERS
X BUS
90
129
MFG. MACH. & PERIPHERAL ID's
FIXED DISK SUPPORT
VIDEO SUPPORT
112
FLOPPY DISK SUPPORT
TPAW-A
UART
PARALLEL PORT
ISA CONTROL
EISA CONTROL
D DATA BUS (ISA SECTION - D15..0)
D DATA BUS (EISA SECTION - D31..16)
SA ADDRESS BUS (ISA ONLY - SA19..0)
LA ADDRESS BUS (ISA SECTION - LA 23..17)
LA ADDRESS BUS (EISA SECTION - LA 31..24)*, LA(16..2)
X-BUS CONTROL SIGNALS
XD DATA BUS BUFFERED D(15..0)
XAIYA ADDRESS BUSES (BUFFERED SA(15..0)/SA(16..1))
ISA CONTROL
FLOPPY DISK LOGIC
FLOPPY CONNECTOR
COM PORT CONNECTOR
PARALLEL CONNECTOR
PRINTER
76
78
80
82
84
86
88
114

FIG. 2B

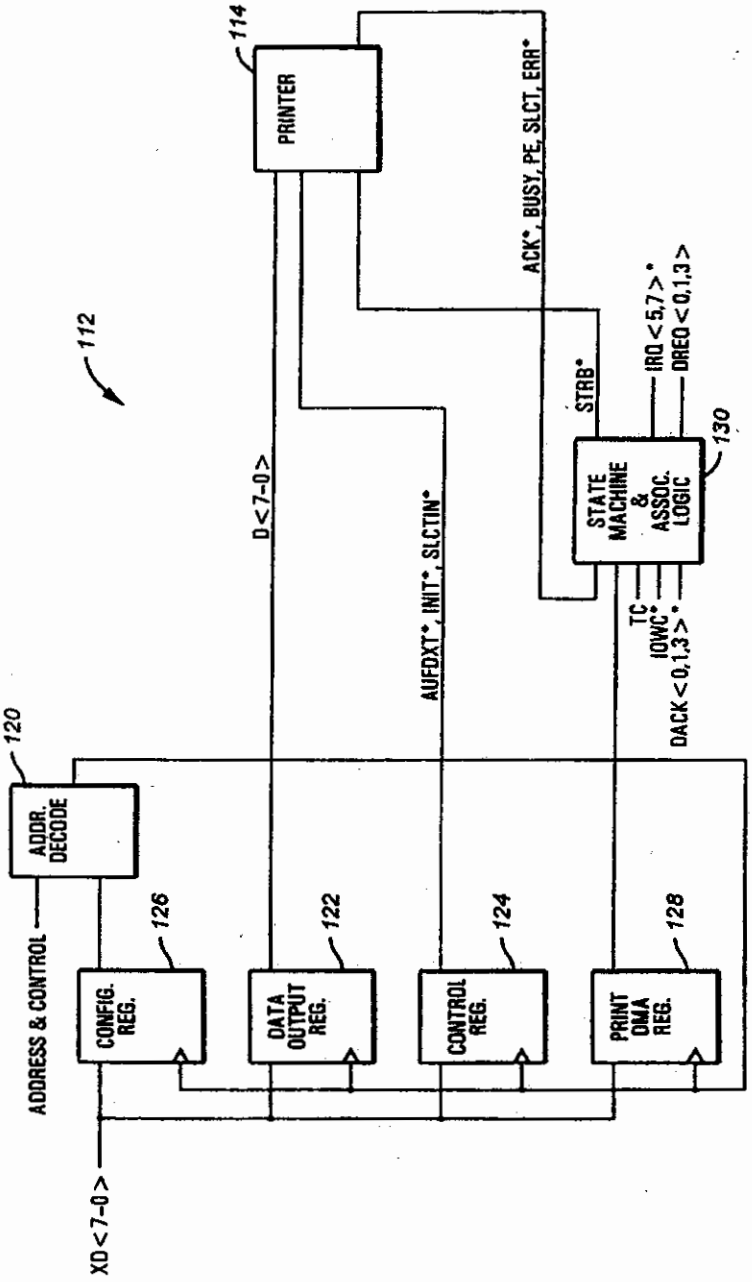

112
114
PRINTER
ACK*, BUSY, PE, SLCT, ERR*
IRQ<5,7>*
DREQ<0,1,3>
STRB*
130
STATE MACHINE & ASSOC. LOGIC
D<7-0>
AUFDXT*, INIT*, SLCTIN*
TC
IOWC*
DACK<0,1,3>*
120
ADDR. DECODE
ADDRESS & CONTROL
126
CONFIG. REG.
122
DATA OUTPUT REG.
124
CONTROL REG.
128
PRINT DMA REG.
XD<7-0>

FIG. 3

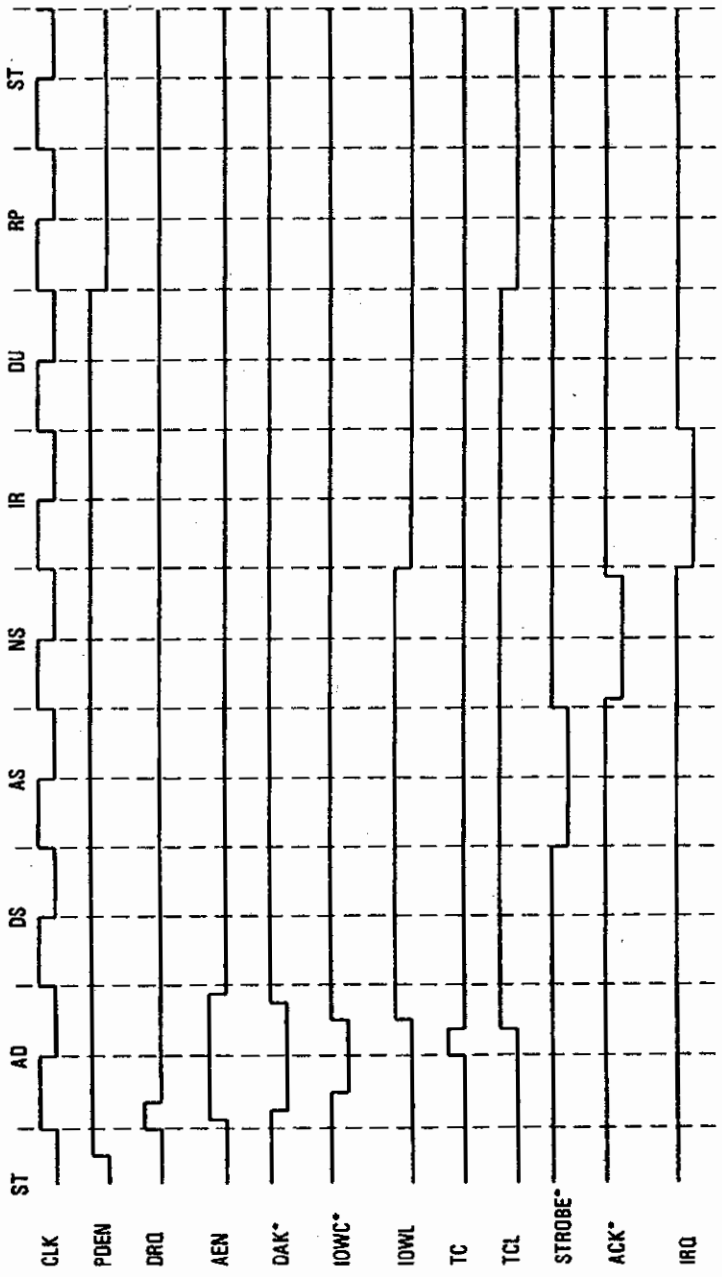

ST
RP
DU
IR
NS
AS
DS
AD
ST
CLK
PDEN
DRQ
AEN
DAK*
IOWC*
IOWL
TC
TCL
STROBE*
ACK*
IRQ

FIG. 4

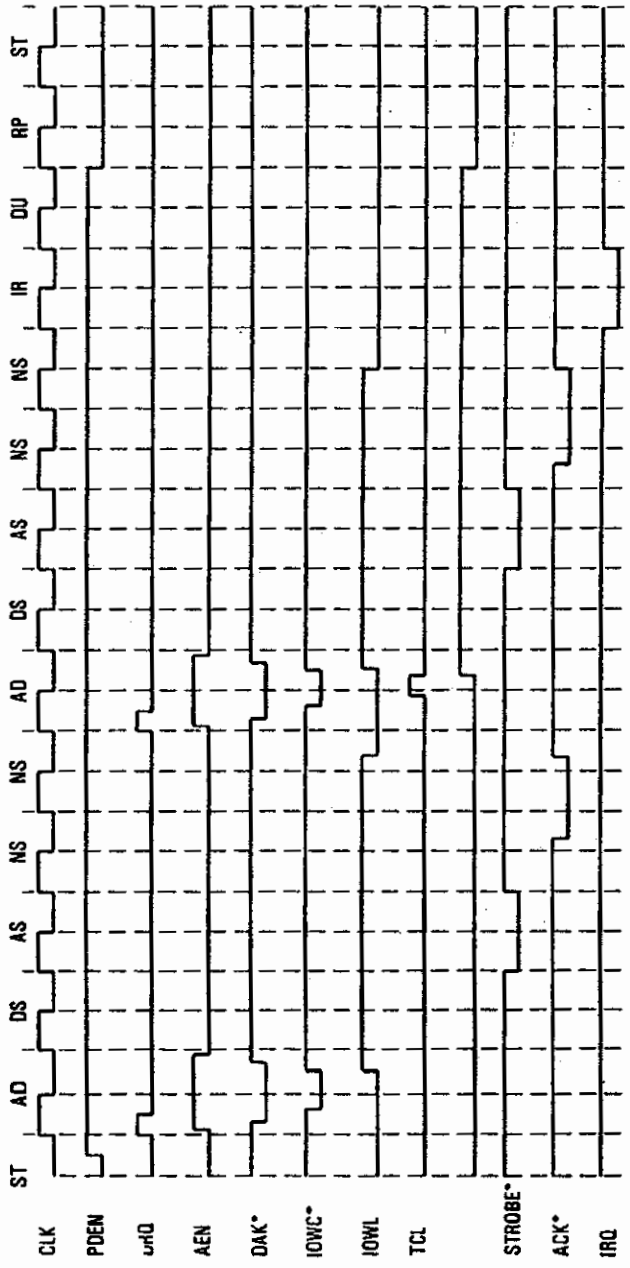

ST
RP
DU
IR
NS
NS
AS
DS
AD
NS
NS
AS
DS
AD
ST
CLK
PDEN
AEN
DAK*
IOWC*
IOWL
TCL
STROBE*
ACK*
IRQ

FIG. 5

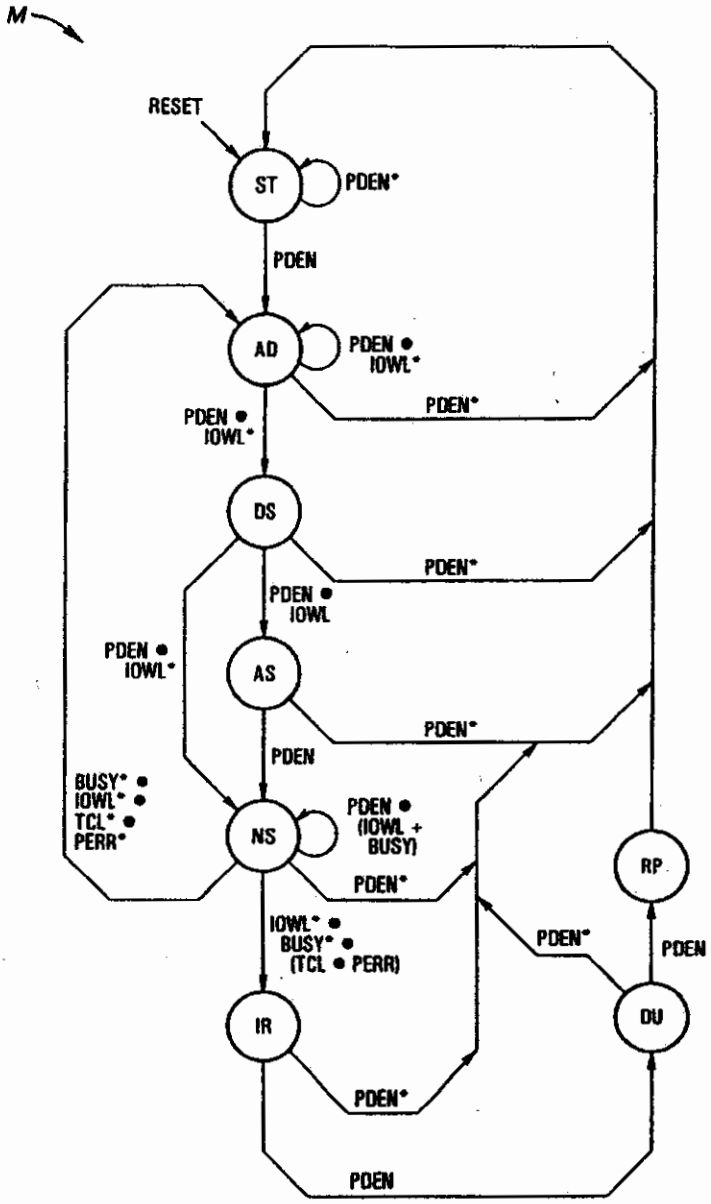

M
RESET
ST
PDEN*
PDEN
AD
PDEN • IOWL*
PDEN*
PDEN • IOWL*
DS
PDEN*
PDEN • IOWL
PDEN • IOWL*
AS
PDEN*
PDEN
BUSY* • IOWL* • TCL* • PERR*
NS
PDEN • (IOWL + BUSY)
PDEN*
RP
IOWL* • BUSY* • (TCL • PERR)
PDEN*
PDEN
IR
DU
PDEN*
PDEN

FIG. 6

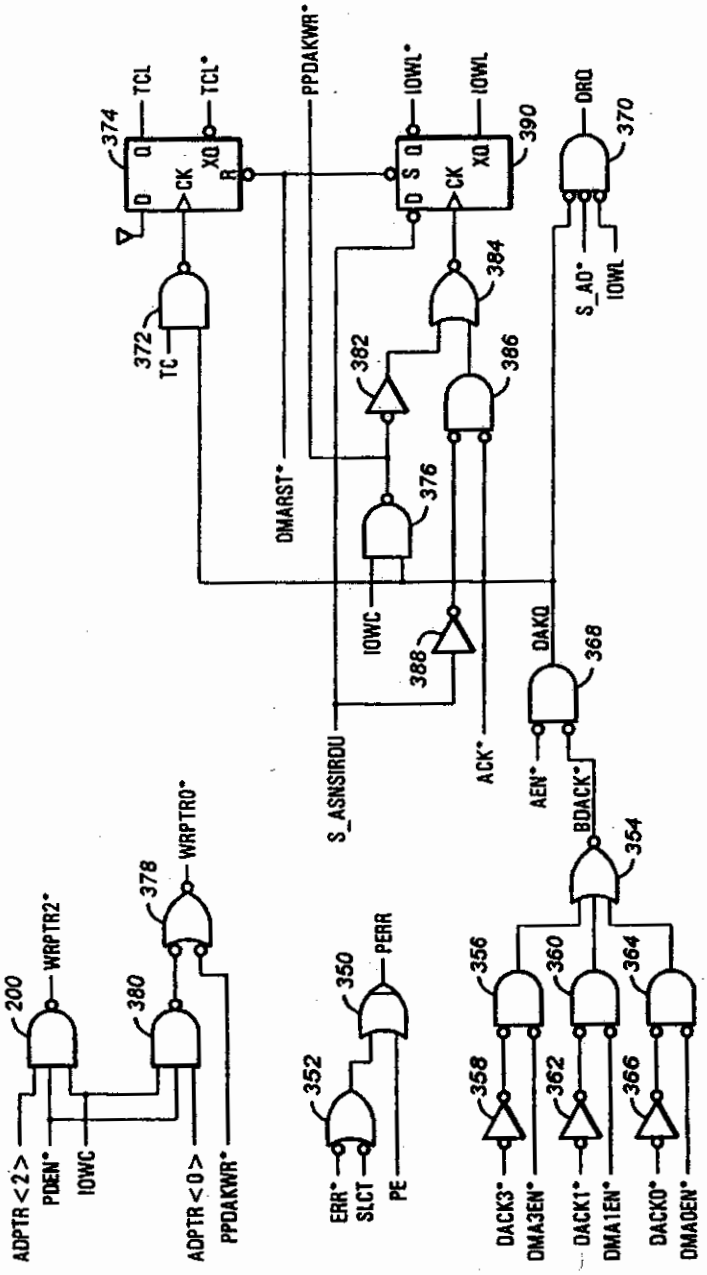

374
TCL
TCL*
PPDAKWR*
IOWL*
IOWL
390
DRQ
370
372
TC
384
382
386
S_AD*
IOWL
DMARST*
376
IOWC
388
DAKQ
368
S_ASNSIRQU
ACK*
AEN*
BDACK*
354
WRPTR0*
378
WRPTR2*
200
380
350
PERR
352
356
360
364
358
362
366
ADPTR<2>
PDEN*
IOWC
ADPTR<0>
PPDAKWR*
ERR*
SLCT
PE
DACK3*
DMA3EN*
DACK1*
DMA1EN*
DACK0*
DMA0EN*

FIG. 7

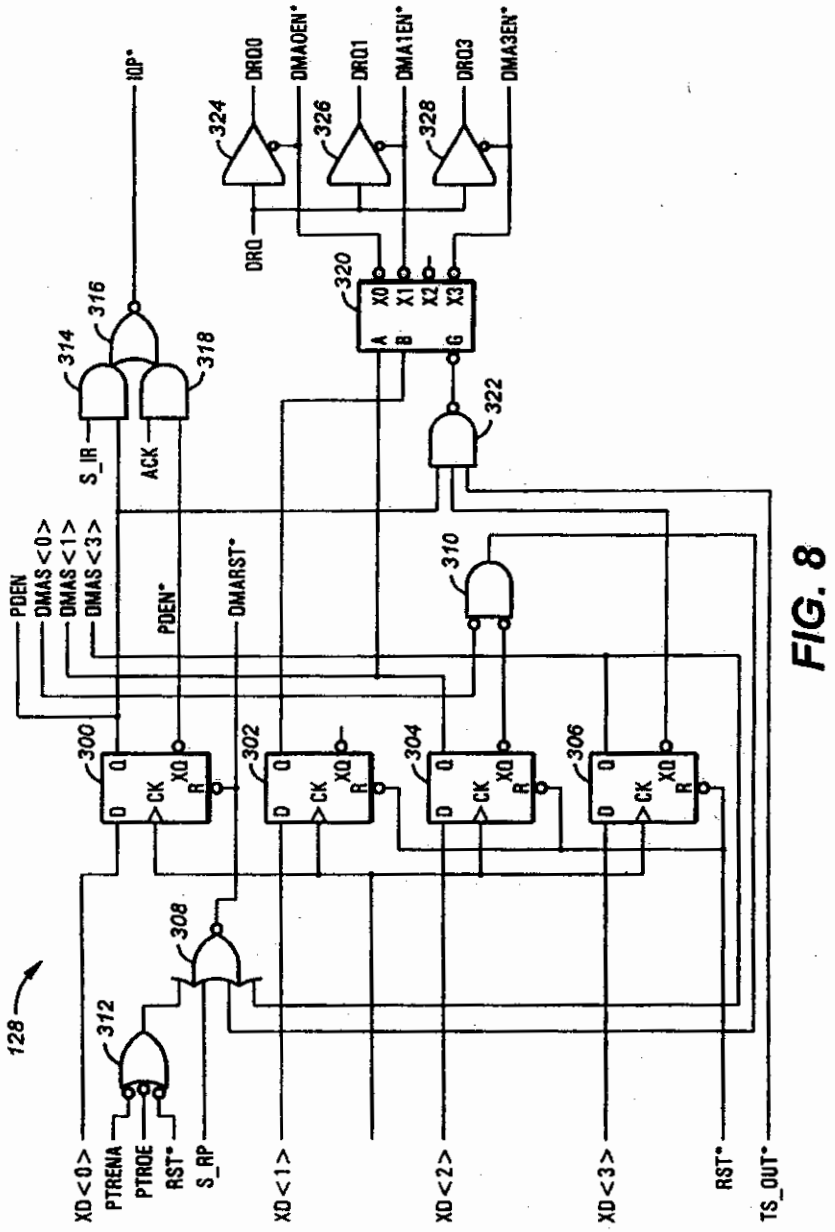

IOP*
DRQ0
DMA0EN*
DRQ1
DMA1EN*
DRQ3
DMA3EN*
324
326
328
DRQ
320
X0
X1
X2
X3
A
B
G
316
314
318
322
S_IR
ACK
PDEN
DMAS<0>
DMAS<1>
DMAS<3>
PDEN*
DMARST*
310
300
302
304
306
D
Q
CK
XQ
R
308
312
128
XD<0>
PTRENA
PTROE
RST*
S_RP
XD<1>
XD<2>
XD<3>
RST*
TS_OUT*

FIG. 8

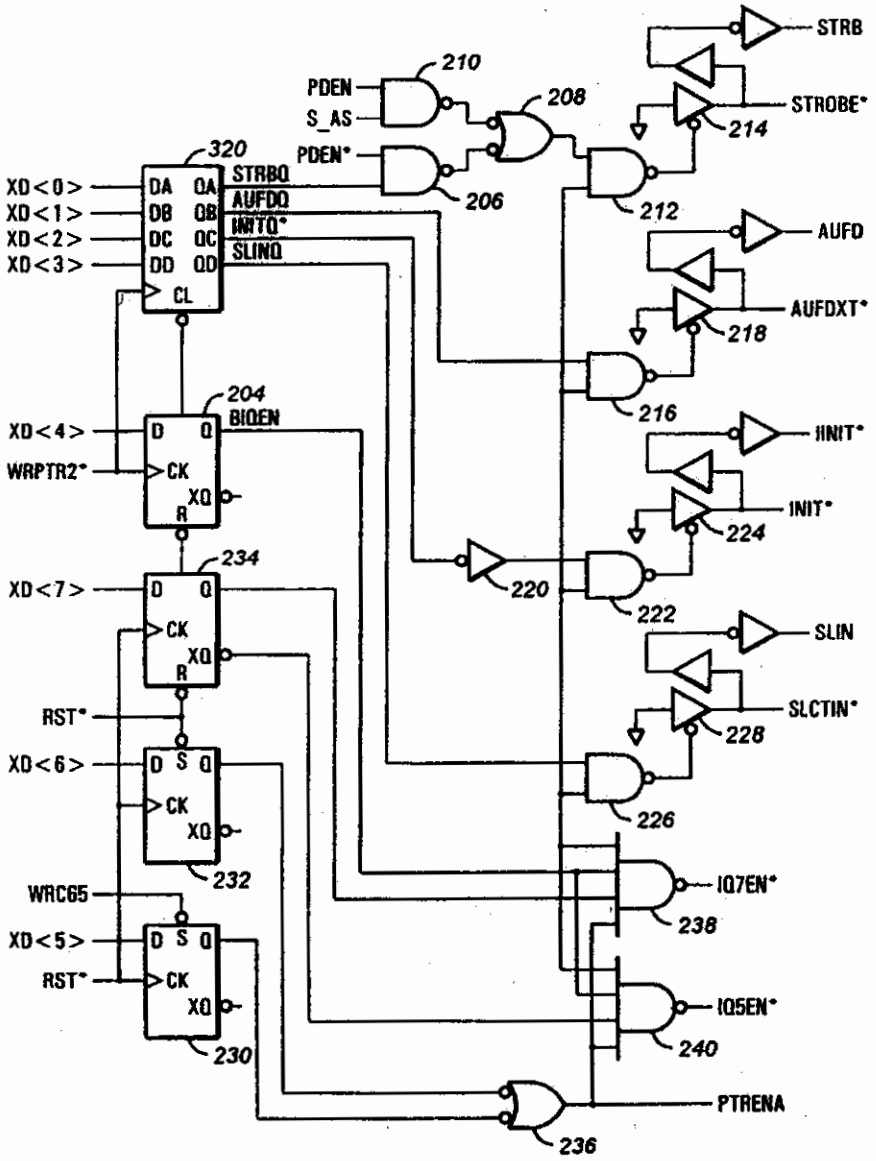

PDEN
S_AS
PDEN*
210
208
206
212
214
STRB
STROBE*
320
STRBQ
AUFDQ
INITQ*
SLINQ
XD<0>
XD<1>
XD<2>
XD<3>
DA OA
DB OB
DC OC
DD OD
CL
216
218
AUFD
AUFDXT*
204
BIOEN
XD<4>
WRPTR2*
D Q
CK
XQ
R
220
222
224
INIT*
INIT*
234
XD<7>
SLIN
SLCTIN*
228
RST*
XD<6>
D S Q
226
232
WRC65
XD<5>
RST*
IQ7EN*
238
IQ5EN*
240
230
PTRENA
236

FIG. 9

SYSTEM FOR PARALLEL PORT WITH DIRECT MEMORY ACCESS CONTROLLER FOR DEVELOPING SIGNAL TO INDICATE PACKET AVAILABLE AND RECEIVING SIGNAL THAT PACKET HAS BEEN ACCEPTED

This is a continuation of U.S. patent application Ser. No. 08/640,223 filed Apr. 30, 1996 U.S. Pat. No. 5,892,976, which is a continuation of U.S. patent application Ser. No. 08/403,585 (now U.S. Pat. No. 5,539,917) filed Mar. 14, 1995, which is a continuation of U.S. patent application Ser. No. 07/431,657 (now abandoned) filed Nov. 3, 1989.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to parallel ports associated with computer systems and with computer systems containing direct memory access capabilities.

2. Description of the Related Art

Personal computer systems are becoming more powerful with each passing moment. Originally they started out with simple 8 bit microprocessors with 1 MHz clock speeds. Microprocessors have now reached capabilities where they are operating with full 32 bit wide data and have clock rates in excess of 30 MHz. Additionally, with these added capabilities the new uses for personal computers have dramatically expanded beyond those originally envisioned. Personal computer systems are now utilized both as powerful work stations performing computer automated engineering functions with high resolution graphic capabilities and as file servers for local area networks. As more sophisticated software is continually developed the capabilities of the microprocessors are stretched so that as much of the time as possible should be applied to the actual processing tasks requested by the user and as little time as possible to various background functions related to controlling the computer system.

Printing with a personal computer compatible with those developed by International Business Machines Corporation (IBM) is done either over an asynchronous serial communications interface or over a parallel interface, commonly referred to as a Centronics type interface. The parallel interface is generally preferred because higher data throughputs are generally available, which decreases actual printer communication time. The parallel port is an 8 bit wide data port with a clocking or strobe signal and various feedback signals indicating data acknowledgement and device busy, as well as printer status signals such as paper out and printer error. This is a relatively simple and straightforward interface and in fact personal computers built according to the standard developed by IBM for its PC and PC/XT may utilize up to three such ports, referred to as LPT1, LPT2 and LPT3 under the MS-DOS environment.

In the past the microprocessor in the personal computer has been required to directly control the parallel port for each individual byte of data being transferred from the computer system to the printer. This resulted in lost computational time in most cases. To transfer a byte of data over the parallel port, the microprocessor first checks the status port for errors and a not busy state. If there are no errors and the printer is not busy, the microprocessor writes the data to the data port. The microprocessor then writes to the status port to set the data strobe signal. The microprocessor then writes to the status port to clear the data strobe signal. This is the minimum number of steps required for each byte of data. If the microprocessor is operating the print task in foreground mode, where the only task it is operating is the printer function and the entire resource of the personal computer is dedicated to monitoring the parallel port, this loop is acceptable, with additional steps needed only to advance in the data file. However, all other user tasks in the processor are on hold until printing is completed. Thus the entire computer is dedicated to the simple task of printing. If however, the printing function is assigned as a background task, then each time a byte must be transferred from the computer system to the printer, an interrupt handling operation must occur. An interrupt occurs, the microprocessor must branch to the printing code described above, the byte is transferred and the microprocessor returns from the interrupt. Now two tasks can be performed basically simultaneously by the microprocessor but even more overhead is added to the print task and relatively large time slices are removed from the foreground task, decreasing the efficiency of both tasks. As can be seen there are a great deal of processor resources involved in transferring each particular byte. This reduces the available microprocessor resources for use by computer programs requested by the user and thus overall system capabilities.

Also present in current personal computer systems is a direct memory access controller. A direct memory access (DMA) controller allows information to be transferred between memory and input/output (I/O) ports without the interaction of the microprocessor. Thus the use of a DMA controller allows the computer system to perform certain functions without requiring an interrupt or dedication of the microprocessor, thus improving computer system capabilities and efficiency. The DMA controller starts an operation after being set up by the microprocessor and then handles the passing of data between the memory and I/O port until the operation is complete, at which time an interrupt is generated by the peripheral device operating in the I/O port space to inform the microprocessor that the operation is complete.

However, this DMA process has never been assigned to the parallel port in personal computers, thus requiring the above mentioned overhead for the printing of each particular byte of data. This overhead can become quite burdensome when large print queues are developed, as when the personal computer is operating as a file server and thus must handle the high throughput requirements of the file server itself as well as developing printing queues and printing tasks for a large number of users.

SUMMARY OF THE INVENTION

The present invention provides the circuitry necessary to interface the DMA controller with the parallel port to allow printing operations to occur without microprocessor intervention once the DMA controller is programmed. The circuitry includes the necessary logic to develop the DMA request and the interrupt signals used to interface with the DMA controller and the microprocessor and produces the strobe signal provided to the printer to indicate to the printer that data is available. The circuitry recognizes the acknowledge, busy, paper out, select and error signals received from the printer and, in combination with the timing signals received from the DMA controller, produces the necessary output signals to transfer data from memory to the printer. Further, the circuitry preferably includes the data register necessary to store the data which is transferred by

the DMA controller from the memory to the I/O port addressed for the particular printer. The circuitry controls the output of the data register to enable the data onto the parallel lines to the printer at the appropriate time when the strobe signal is being presented.

A state machine is used to clock the logic through various states depending upon the present state and given conditions and, with associated combinatorial logic, produces the necessary output signals to inform the printer that it has new data to acknowledge, to inform the DMA controller that another byte of data is requested and to indicate to the microprocessor by means of an interrupt line that either the task is completed or errors have occurred.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained when the following detailed description of the preferred embodiment is considered in conjunction with the following drawings, in which:

FIG. 1, FIG. 2A and FIG. 2B are schematic block diagrams of a computer system incorporating the present invention;

FIG. 3 is a more detailed schematic block diagram of circuitry incorporating the present invention;

FIGS. 4 and 5 are timing diagrams for various signals utilized in operation of the present invention;

FIG. 6 is a state machine diagram of operating sequences of portions of the circuitry of FIG. 3; and

FIGS. 7, 8 and 9 are detailed schematic diagrams of circuitry associated with the state machine of FIG. 6 and with the block diagram of FIG. 3.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to FIG. 1, FIG. 2A and FIG. 2B, the letter C designates generally a computer system incorporating the present invention. For clarity, system C is shown in two portions, with the interconnections between FIG. 1, FIG. 2A and FIG. 2B designated by reference to the circled numbers one to eight. System C is comprised of a number of block elements interconnected via four buses. Throughout this specification, signal mnemonics with an asterisk following the signal descriptors indicates the signal is active at a logic low level. Signal mnemonics having numbers or ranges between angled brackets refer to those particular bits or positions in a bus.

In FIG. 1, a computer system is depicted. A central processing unit CPU comprises a processor 20, a numerical coprocessor 22 and a cache memory controller 24 and associated logic circuits connected to a local processor bus 26. Associated with cache controller 24 is high speed cache data random access memory 28, noncacheable memory address map programming logic circuitry 30, noncacheable address memory 32, address exchange latch circuitry 34 and data exchange transceiver 36. Associated with the CPU also are local bus ready logic circuit 38, next address enable logic circuit 40 and bus request logic circuit 42.

The processor 20 is preferably an Intel 80386 microprocessor. The processor 20 has its control, address and data lines interfaced to the local processor bus 26. The coprocessor 22 is preferably an Intel 80387 and/or Weitek WTL 3167 numeric coprocessor interfacing with the local processor bus 26 and the processor 20 in the conventional manner. The cache ram 28 is preferably suitable high-speed static random access memory which interfaces with the address and data elements of bus 26 under control of the cache controller 24 to carry out required cache memory operations. The cache controller 24 is preferably an Intel 82385 cache controller configured to operate in two-way set associative master mode. In the preferred embodiment the components are the 33 MHz versions of the respective units. Address latch circuitry 34 and data transceiver 36 interface the cache controller 24 with the processor 20 and provide a local bus interface between the local processor bus 26 and a host bus 44.

Circuit 38 is a logic circuit which provides a bus ready signal to control access to the local bus 26 and indicate when the next cycle can begin. The enable circuit 40 is utilized to indicate that the next address of data or code to be utilized by subsystem elements in pipelined address mode can be placed on the local bus 26.

Noncacheable memory address map programmer 30 cooperates with the processor 20 and the noncacheable address memory 32 to map noncacheable memory locations. The noncacheable address memory 32 is utilized to designate areas of system memory that are noncacheable to avoid many types of cache memory incoherency. The bus request logic circuit 42 is utilized by the processor 20 and associated elements to request access to the host bus 44 in situations such as when requested data is not located in the cache memory 28 and access to system memory is required.

In the drawings, system C is configured having the processor bus 26, the host bus 44, an extended industry standard architecture (EISA) bus 46 (FIG. 2A and FIG. 2B) and an X bus 90. The details of the portion of the system illustrated in FIG. 2A and FIG. 2B, and not discussed in detail below are not significant to the present invention other than to illustrate an example of a fully configured computer system. The EISA specification Version 3.1 is provided in U.S. Pat. No. 5,101,492 filed Sep. 3, 1989, issued Mar. 31, 1992, and entitled "Data Redundancy and Recovery Protection" by Schultz, et al., and fully explains the requirements of an EISA system and is hereby incorporated by reference. The portion of system C illustrated in FIG. 2A and FIG. 2B is essentially a configured EISA system which includes the necessary EISA bus 46, and EISA bus controller 48, data latches and transceivers 50 and address latches and buffers 52 to interface between the EISA bus 46 and the host bus 44. Also illustrated in FIG. 2A and FIG. 2B is an integrated system peripheral 54, which incorporates a number of the elements used in an EISA-based computer system.

The integrated system peripheral (ISP) 54 includes a direct memory access controller 56 for controlling access to main memory 58 (FIG. 1) or memory contained in EISA slots and input/output (I/O) locations without the need for access to the processor 20. The main memory array 58 is considered to be local memory and comprises a memory circuit array of size suitable to accommodate the particular requirements of the system. The ISP 54 also includes interrupt controllers 70, nonmaskable interrupt logic 72 and system timers 74 which allow control of interrupt signals and generate necessary timing signals and wait states in a manner according to the EISA specification and conventional practice. In the preferred embodiment, processor generated interrupt requests are controlled via dual interrupt control circuits emulating and extending conventional Intel 8259 interrupt controllers. The ISP 54 also includes bus arbitration logic 75 which, in cooperation with the bus controller 48, controls and arbitrates among the various requests for the EISA bus 46 by the cache controller 24, the DMA controller 56 and bus master devices located on the EISA bus 46.

The main memory array **58** is preferably dynamic random access memory. Memory **58** interfaces with the host bus **44** via a data buffer circuit **60**, a memory controller circuit **62** and a memory mapper **68**. The buffer **60** performs data transceiving and parity generating and checking functions. The memory controller **62** and memory mapper **68** interface with the memory **58** via address multiplexer and column address strobe buffers **66** and row address enable logic circuit **64**.

The EISA bus **46** includes ISA and EISA control buses **76** and **78**, ISA and EISA control buses **80** and **82** and address buses **84**, **86** and **88**. System peripherals are interfaced via the X bus **90** in combination with the ISA control bus **76** from the EISA bus **46**. Control and data/address transfer for the X bus **90** are facilitated by X bus control logic **92**, data transceivers **94** and address latches **96**.

Attached to the X bus **90** are various peripheral devices such as keyboard/mouse controller **98** which interfaces the X bus **90** with a suitable keyboard and mouse via connectors **100** and **102**, respectively. Also attached to the X bus **90** are read only memory circuits **106** which contain basic operations software for the system C and for system video operations. A serial communications port **108** is also connected to the system C via the X bus **90**. Floppy and fixed disk support, a parallel port **112** generally used to communicate with the printer **114**, a second serial port, and video support circuits are provided in block circuit **110**.

The parallel port **112** is shown in more detail in FIG. **3**. The parallel port **112** includes address decode circuitry **120** which decodes the various addresses provided on the X bus **90** for addresses associated with the particular printer port selected or selected control registers. In the preferred embodiment the printer port I/O port address blocks are 3BC–3BF for LPT1, 378-37B for LPT2, and 278-27B for LPT3, these being the conventional hexadecimal addresses according to the standards developed for IBM PC compatible personal computers. The preferred addresses of the various control registers will be noted with the register description.

A data output register **122** is connected to the data bus XD **129** to receive the lower 8 bits of data and is connected to the address decode circuitry **120** to properly clock in the data. This data output register **122** is that commonly associated with the parallel port at the base or zero bias address of the particular address block. A control register **124** is also connected to the data bus XD **130** and receives a clocking signal from the address decode circuitry **120**, preferably at the second address of the particular address block, to provide the auto linefeed, initialize printer and select in signals associated with the printer **114** in the conventional interface. A configuration register **126**, preferably located at I/O port address 0C65, is provided to allow selection of the particular parallel port address block to be associated with the parallel port **112** and the interrupt to be utilized by the port. Preferably interrupts IRQ5 or IRQ7 are utilized, with the defaults for the port being LPT1 and IRQ7.

An additional register referred to as the printer DMA register **128** is provided in a computer system incorporating the present invention to allow control of the direct memory access parallel port option. The printer DMA register **128** is also connected to the data bus XD **129** and to the address decode circuitry **120**. This register **128** is preferably located at I/O port address 0C7B and is a read/write port. Preferably 3 bits are allocated for determining the desired DMA channel and 1 bit is provided to enable or disable parallel port DMA operation and to indicate status in the cycle of the circuitry. A state machine and associated logic circuitry **130** is provided to control operation so that the appropriate signals are produced to the printer **114**, the interrupt controller **70** and the DMA controller **56** based on the signals provided from the DMA controller **56**, the printer **114** and the printer DMA register **128**. Details of the state machine and associated logic **130** will be explained in greater detail.

Referring now to FIG. **4**, a timing diagram is provided illustrating the operation of a single cycle or transfer of a single byte of data from memory to the printer **114**. The basic clocking signal for the operation is the CLK signal, which is preferably a signal having frequency less than 2 MHz, 1.8432 MHz in the preferred embodiment. This frequency limit is preferred because the STROBE* signal presented to the printer **114** must be at least 500 nsec long according to the general specifications and therefore the use of a 2 MHz or lower clock frequency readily provides this pulse width. The CLK signal is used to clock the state machine through of the various states, which are indicated above the particular clock phase in FIG. **4**. The parallel port DMA enable signal, referred to as PDEN, is used to trigger the operation. When the PDEN signal goes high this is an indication that data is to be transferred to the parallel port under the control of the DMA controller **56**. Then on the next rising edge of the CLK signal the DRQ or DMA request signal for the selected DMA channel is raised to indicate to the DMA controller **56** that a DMA request is desired and information should be passed. The DMA controller **56** acknowledges the request by gaining control of the bus and raising the AEN signal, to indicate that a memory address is being presented in the address bus and that any I/O devices except the acknowledged DMA device are to ignore the address. Additionally, the DMA controller **56** lowers the appropriate DAK* or DMA acknowledge signal to indicate that the DMA request is acknowledged. Upon seeing this acknowledgement of its DMA request, the circuitry **130** lowers the DREQ line so that only a single byte is transferred. The DMA controller **56** then proceeds along its course and produces the IOWC* signal, which indicates that data is being read from the memory and is to be written to the I/O port location requesting the information at this time. Additionally, the DMA controller **56** indicates that this is the last or terminal count byte to be transferred in the particular case of FIG. **4** by setting the TC signal high. on the falling edge of the TC signal a signal referred to as TCL is provided high to indicate to the circuitry **130** that a terminal count has been reached. On the rising edge of the IOWC* signal, a signal referred to as IOWL is asserted or made high to indicate to the circuitry **130** that data has been presented to the circuitry **130** for presentation to the printer **114**.

The circuitry **130** proceeds through several states of the state machine until it reaches state AS, at which time the STROBE* signal goes low to indicate to the printer **114** that data is present on the data lines. On the next rising edge of the CLK signal the STROBE* signal is raised, thus strobing the data into the printer **114**. The printer **114** responds by lowering the ACK* signal to acknowledge the receipt of the data. On the next rising edge of the CLK signal following the receipt of the ACK* signal, the IOWL signal is lowered, thus clearing the indication to the circuitry **130** of a need to transfer data. Also at this time the IRQ or interrupt request signal goes low to indicate that the desired data transfer has been completed, and therefore the printer operation is complete. The IRQ signal is held low for one CLK cycle. On the next rising edge of the CLK signal the TCL signal is lowered, thus clearing the terminal count indication in the circuitry **130**. At this time the state machine automatically

lowers the PDEN signal to indicate completion of the cycle and to allow another cycle to be initiated. The state machine then proceeds to a resting state ST.

FIG. 5 shows a condition where 2 bytes of data are passed from to the printer 114 by the DMA controller. The PDEN signal undergoes a rising edge to initiate the process. The DREQ signal is then asserted to indicate the request to transfer a byte of data to the printer 114. The DAK* and IOWC* signals are then asserted by the DMA controller 56 and the data is transferred to the circuitry 130. The STROBE* signal is produced as before to strobe the data into the printer 114. In this case however the printer 114 is slightly slower in responding and the ACK* signal does not return to a high level until after the next rising edge of the CLK signal following the removal of the STROBE* signal, thus holding the IOWL signal high past a CLK signal rising edge. This is considered a wait state condition in relation to the state machine and therefore an additional CLK cycle is used in transferring the data. When the ACK* signal is raised the IOWL signal goes low, while the TCL signal remains low and the PDEN signal remains high, indicating that more information is to be transferred. Thus the DREQ signal is reasserted to indicate to the DMA controller that a second byte of data should be transferred. This then initiates a second cycle transferring the second byte, with the timing of this particular transfer being similar to the timing in FIG. 4, in that it is a terminal transfer and the TC signal is raised by the DMA controller 56.

As indicated, a state machine M (FIG. 6) controls operation of the circuitry 130. Upon reset of the computer system C the state machine M starts in state ST, referred to as the start or reset state. The state machine M stays in state ST while the PDEN signal is low, indicating that the printer DMA transfer has not been enabled. When the PDEN signal goes high, indicating that a transfer is desired, control proceeds on the next rising edge of the CLK signal to state AD, the accept data state. Control remains in state AD while the PDEN signal is high and the IOWL* signal is high, indicating that the IOWC* signal strobe has not been completed. This condition is possible because the memory devices responding to the request may be slow and thus need wait states. If the PDEN signal goes low, indicating that a request is not necessary, control transfers from state AD to state ST. If however, the PDEN signal is high and the IOWL signal goes high, indicating that the data has been transferred from the memory into the data output register 122, control proceeds to state DS on the rising edge of the CLK signal. The IOWL signal thus indicates that data is present and needs to be transferred from the circuitry 130.

From state DS, the data setup state, control proceeds to state ST if the PDEN signal is the lowered. If for some reason the IOWL signal should be removed while the parallel DMA function is enabled, control proceeds to state DS to state NS, the negate strobe state. In this case it is noted that no STROBE* signal is provided to the printer 114. Under normal conditions the PDEN signal is high and the IOWL signal is high, so control proceeds from state DS to state AS, the assert strobe state. If the PDEN* signal should be low at the next rising edge of the CLK signal, control proceeds to state ST. If the PDEN signal remains high, then control proceeds to state NS. Control remains in state NS until the printer has responded to the data that has been strobed. This is indicated by the PDEN signal being high and either the IOWL signal being high, which indicates an acknowledgement has not been completed, or the BUSY signal, which is received from the printer 114 to indicate that it is busy, being high. If the PDEN signal should be removed, control returns to state ST. When the printer 114 acknowledges that it has received the information so that the IOWL signal goes low and the BUSY signal goes low, then control proceeds from state NS. If this was not the last byte to be transferred and no printer errors were received, then control proceeds from state NS to state AD to receive the next byte of data. If, however, this was the last byte of data to be transferred or a printer error occurred, when the printer 114 acknowledges completion of the particular byte transfer, control proceeds from state NS to state IR, the interrupt request state. Control proceeds from state IR to state ST if the PDEN signal goes low and proceeds to state DU, a dummy state, if the PDEN signal remains high, indicating that the operation is to continue. From state DU control proceeds to state ST if the PDEN signal is removed or to state RP if the PDEN signal is still asserted. In state RP, the reset state, the PDEN signal is cleared and control proceeds to state ST in all cases. The exact details of the construction of the state machine M are not shown as this is considered to be within the level of those skilled in the art.

Referring now to FIG. 7, various associated logic circuitry is necessary in combination with the state machine M to perform the functions of the circuitry 130. When the parallel port DMA function is enabled, it is preferably desired that the processor 20 cannot directly access the parallel control port or the parallel data port, thus interfering with the DMA controller-based print function. To this end a three input NAND gate 200 receives inputs of the IOWC signal, to indicate that a address strobe is present; the PDEN* signal, which indicates the status of the printer DMA function; and the ADPTR<2> signal, which is an address decode signal indicating that an I/O operation is being requested at the parallel control port location. Thus when the PDEN signal is high, indicating that the printer DMA function is enabled, the write pulse to the parallel control port from the processor 20 is disabled. The output of the NAND gate 200 is the WRPTR2* signal which is provided as a clocking input to a 4 bit D-type flip-flop 202 (FIG. 9) and a single bit D-type flip-flop 204. These two flip-flops 202 and 204 comprise the control register 124 and are appropriately connected to the XD data bus 129 to receive the bit positions designated to control the strobe output, the auto linefeed output, the initialize output, the select output, and the interrupt enable output.

The bit 0 position corresponds to the strobe output of the flip-flop 202, which is presented to one input of a two input NAND gate 206. The other input of the NAND gate 206 is the PDEN* signal, so that the output of the NAND gate 206 is disabled if the printer DMA function is enabled. The output of the NAND gate 206 is provided to one input of a two input NAND gate 208, whose other input is received from the output of a two input NAND gate 210. The inputs to the two input NAND gate 210 are the PDEN signal and a signal referred to as S_AS, which indicates that the state machine M is in state AS, the address strobe state. The output of NAND gate 208 is provided as one input to a two input NAND gate 212, whose output is connected to the inverted tri-state control input of an output buffer 214. The input to the buffer 214 is connected to ground and the output is the STROBE* signal provided to the printer 114. The second input to the NAND gate 212 is a signal referred to as TS_OUT*, which allows the parallel outputs to be disabled if desired.

The bit 1 position of the XD data bus 129 corresponds to the auto linefeed signal. This particular output of the flip-flop 202 is connected to one input of a two input NAND gate 216, whose other input receives the TS_OUT* signal. The

output of the NAND gate **216** is connected to the inverted tri-state control input of a buffer **218**, whose input is connected to ground and whose output represents the AUFDXT* or auto linefeed signal.

The bit 2 position of the data bus corresponds to the initialize printer signal and the appropriate output of the flip-flop **202** is presented to an inverter **220** whose output is connected to one input of a NAND gate **222**, whose other input receives the TS_OUT* signal. The output of the NAND gate **222** is connected to the inverted tri-state control input of a buffer **224**, whose input is grounded and whose output is the INIT* signal to cause the printer **114** to be initialized.

The bit 3 position of the data bus corresponds to the select line provided to the printer and the output of the flip-flop **202** is provided to one input of a NAND gate **226**, whose other input receives the TS_OUT* signal. The output of the NAND gate **226** is provided to the inverted tri-state control input of a buffer **228**, whose input is grounded. The output of buffer **228** is the SLCTIN* signal, which when asserted low, indicates that the printer **114** is selected.

The parallel configuration register **126** is preferably located at address 0C65 and contains 2 bits to select the desired parallel port address block to be utilized and 1 bit to determine which interrupt is to be used, as previously mentioned. If the 2 bits utilized to select the parallel port address block are both in a high state, this is an indication that the parallel port is disabled in the preferred embodiment. A signal referred to as WRC65, which indicates a write strobe to I/O port 0C65, the preferred I/O port address of the parallel control register **126**, is provided to the clocking inputs of 3 D-type flip-flops **230**, **232** and **234**. The D inputs of the flip-flops **230** and **232** are connected respectively to bits 5 and 6 of the data bus XD **130** and are used to select the parallel port address block to be utilized. The noninverted outputs of the flip-flops **230** and **232** are connected as the two inputs to a two input NAND gate **236**, whose output is the PTRENA or printer enable signal. The D input to the flip-flop **234** is connected to bit position 7 of the XD data bus **130** and is used to indicate which interrupt line is to be utilized. The noninverted output of the flip-flop **234** is provided as one input to a 4 input NAND gate **238**, whose other inputs are the TS_OUT* signal, the signal present at the noninverting output of the flip-flop **204**, which indicates that the interrupts are enabled, and the PTRENA signal, which indicates that the printer **114** is enabled. The output of the NAND gate **238** is the IQ7EN* signal which, when asserted low, indicates that IRQ7 is to be utilized. The inverted output of the flip-flop **234** is provided along with the BIQEN signal provided as the output of flip-flop **204**, the TS_OUT* signal and the PTRENA signal to a four input NAND gate **240**, whose output is the IQ5EN* signal which indicates, when low, that IRQ5 is to be utilized for the parallel port.

The printer DMA register **128** is shown in FIG. **8**. This register **128** has an address of 0C7B in the preferred embodiment. A clocking strobe referred to as WRXC7B* is provided to the clocking inputs of four D-type flip-flops **300**, **302**, **304** and **306**, which form the 4 bits of the register **128**. The inverted reset inputs of the flip-flops **302**, **304** and **306** are connected to the RST* signal, which is a signal that indicates that the computer system C is being reset. Bit positions 1, 2 and 3 of the data bus XD are connected respectively to the D inputs of the flip-flops **302**, **304** and **306** to indicate which DMA channel is selected. The non inverted output of the flip-flop **302** is the DMAS<0> signal, the noninverted output of the flip-flop **304** is the DMAS<1> signal and the noninverted output of the flip-flop **306** is the DMAS<2> signal. These three signals are the encoded value of the particular DMA channel to be utilized. While 3 bits are provided for DMA channel selection, in the preferred embodiment only 3 of the 8 possible channels can be utilized to prevent interference with other devices which utilize the DMA controller **56**. The allowed channels in the preferred embodiment are DMA channels 0, 1 and 3. If the register **136** is programmed to utilize channels 2 or 4–7, then the printer DMA function is automatically disabled. This is provided by connecting the noninverted output of the flip-flop **306** to one input of a four input NOR gate **308**, whose output is connected to the inverted reset input of the flip-flop **300**, whose noninverted output is the PDEN signal. Additionally, the noninverted output of the flip-flop **302** and the inverted output of the flip-flop **304** are provided as the two inputs to a two input NOR gate **310**, whose output is also one of the inputs to the NOR gate **308**. A third input to the NOR gate **308** is the S_RP signal, which indicates that the state machine M is in state RP, the reset state. The fourth input to the NOR gate **308** is provided by the output of a three input NAND gate **312**, whose inputs are the PTRENA signal, the PTROE signal, which indicates that the printer output function is enabled, and the RST* signal. The output of the NOR gate **308** is referred to as the DMARST* signal to indicate it is for resetting of the printer DMA function. The 0 bit position of the XD data bus **130** is connected to the D input of the flip-flop **300** to allow the computer C to turn on or enable the printer DMA function. Additionally, the four outputs of the flip-flops **300**, **302**, **304** and **306** are provided to the data bus XD **130** through circuitry (not shown) allowing the computer system C to read the particular values present in the register **128**, thus making this a full read/write register.

The PDEN signal is provided as one input to a two input AND gate **314**, whose other input is the S_IR signal, which indicates that the state machine M is in state IR. This AND gate **314** is used to provide the IRQ pulse to the proper interrupt request line when the printer DMA function is enabled. The output of the AND gate **314** is connected to one input of a two input NOR gate **316**, whose output is the IQP* signal. The other input of the NOR gate **316** is connected to the output of a two input AND gate **318**, whose inputs receive the PDEN* signal and the ACK signal. This AND gate **318** is utilized when the printer DMA function is not enabled so that an interrupt is generated whenever the printer **114** acknowledges the receipt of the data.

Selection of the particular DMA channel which is to receive the DMA request signal is provided by the use of a 2–44 decoder **320**. The noninverted outputs of the flip-flop **302** and the flip-flop **304** are provided to the selection inputs of the decoder **320**, while the output of a three input NAND gate **322** is provided to the inverted enable input of the decoder **320**. The three inputs to the NAND gate **322** are the PDEN signal, the inverted output of the flip-flop **306** and the TS_OUT* signal. Thus the DMA request function is disabled when either the printer DMA function is disabled, the outputs are tristated or one of the upper four DMA channels are selected. The decoder **320** produces a low signal based on the binary encoded value of the two input signals, so that the 0 output corresponds to a requested DMA channel 0, this being the DMA0EN* signal, which is connected to the inverted tristate control input of a buffer **324**, whose input is the DRQ signal and whose output is the DRQ0 signal. The output of the 1 position of the decoder **320** is the DMA1EN* signal and is provided to the inverted tristate control input of a buffer **326**, whose input receives the DRQ signal and

whose output is the DRQ1 signal. Similarly, the bit 3 position of the decoder **320** is the DMA3EN* signal and is provided to the inverted tristate control input of a buffer **328** whose input receives the DRQ signal and whose output is the DRQ3 signal. DRQ0, DRQ1 and DRQ3 outputs are thus provided to the XD bus **90** as appropriate to allow the DMA controller **56** to be alerted.

As noted while discussing the state machine M the state machine M branches according to certain signals. One of these signals is the PERR or printer error signal, which is produced as the output of a two input OR gate **350** (FIG. 7) one of whose inputs is the PE signal, which is the paper end signal from the printer **114**, and whose other input is the output of a two input NAND gate **352**, which receives the ERR* signal, which indicates that there has been a printer error, and the SLCT signal, which indicates that the printer **114** is selected. Thus when either the printer **114** is out of paper, there is a printer error or the printer **114** has been deselected, the PERR signal goes high to indicate this error condition.

Various logic is also necessary to receive the various DMA acknowledge signals. A signal referred to as BDACK*, a buffered DMA acknowledge signal, is produced as an output of a three input NOR gate **354**. One of the inputs of the NOR gate **354** is the output of a two input NOR gate **356** whose inputs are the DMA3EN* signal and the output of a buffer **358** which is connected to the DACK3* signal provided on the X bus **90**. A second input to the NOR gate **354** is provided by the output of a two input NOR gate **358**, one of whose input signals is the DMA1EN* signal and whose other input is received from a buffer **362** whose input is the DACK1* signal. Similarly, the third input of the NOR gate **354** is provided by the output of a two input NOR gate **364**, one of whose inputs is the DMA0EN* signal and whose other input is provided by the output of a buffer **366**, which is connected to the DACK0* signal. Thus the BDACK* signal indicates the presence of a DMA acknowledge on the appropriate DMA channel as selected by the printer DMA control register.

The BDACK* signal is provided as a one input to a two input NOR gate **368** whose other input receives the AEN* signal. The output of the NOR gate **368** is the DAKQ signal, which indicates that a valid DMA acknowledge has been received. This output signal is provided as one input to a three input NOR gate **370** whose other inputs are the IOWL signal and the S_AD* signal, which indicates that the state machine M is not in state AD. The output of the NOR gate **370** is the DRQ signal which is provided to the buffers **324**, **326** and **328** for presentation to the X bus **90**. Thus the DRQ signal is presented whenever the state machine enters state AD, a DMA acknowledgement on the particular channel is not present and the IOWL signal is not active.

The DAKQ signal is also provided as one input to a two input NAND gate **372** whose other input is the TC signal from the X bus **90**. The output of NAND gate **372** is provided as the clocking input to a D-type flip-flop **374**, whose D input is connected to a high logic level and whose output is the TCL signal. The inverted reset input to the flip-flop **374** is connected to the DMARST* signal. Thus whenever a proper DMA cycle is in progress, as indicated by DAKQ signal and the TC signal is received, then on the following edge of the TC signal the TCL signal is generated. The TCL signal is cleared when the state machine reaches the reset state RP.

The DAKQ signal is also provided as one input to a two input NAND gate **376**, whose other input is the IOWC signal. The output of the NAND gate **376** is referred to as the PPDAKWR* signal, that is, the parallel port DMA acknowledge write strobe, to indicate that data is to be clocked into the parallel port data output register **122**. The signal is provided as one input to a two input AND gate **378**, whose other input is connected to the output of a three input NAND gate **380**. The inputs of the NAND gate **380** are the PDEN* signal, the IOWC singal and the ADPTR<0> signal, which is the address decode to indicate that the parallel port data output register **122** is being addressed by the computer. Thus the output of the NAND gate **380** is disabled when the parallel DMA function is activated and in those cases the data output register **122** clocking strobe is provided by the PPDAKWR* signal through the AND gate **378**.

The PPDAKWR* signal is also provided to an inverter **382**, whose output is connected to one input of a two input NOR gate **384**. The other input of the NOR gate **384** is provided by the output of a two input NOR gate **386**, one of whose inputs is the ACK* signal and the other is the output of an inverter **388**. The input to the inverter **388** is a signal which indicates that the state machine is in one of the states AS, NS, IR or DU. The S_ASNSIRDU signal is also provided to the inverted D input of a D-type flip-flop **390**. The output of the NOR gate **384** is provided to the clocking input of the flip-flop **390**, while the DMARST* signal is provided to the inverted set input of the flip-flop. The inverted output of the flip-flop **390** is the IOWL* signal, while the noninverted output of the flip-flop **390** is the IOWL signal. Thus the IOWL signal is developed on the rising edge of the IOWC* strobe when a printer DMA cycle is in progress and is cleared either when state machine M enters state RP or when the printer acknowledges the receipt of data, generally in state NS.

The parallel port or printer DMA function is operated using only a few steps which use a minimal amount of the processor's time. The data to be printed is located in a continuous location in memory. This can be as a result of the user's program or after placement by a function call. The starting address and number of bytes to be transferred is known. The desired channel of the DMA controller **56** is set up for a demand mode transfer with the initial memory address and byte count loaded into the DMA controller **56**. The appropriate DMA channel is set in the printer DMA register **128**. A one value is written to the PDEN flip-flop **300** in the printer DMA register **128** and the transfer sequence commences. When the transfer is completed the processor **20** is interrupted and any necessary completion tasks are performed. Thus large data blocks can be printed without processor control of each byte, increasing overall efficiency of the computer system C.

The foregoing disclosure and description of the invention are illustrative and explanatory thereof, and various changes in the size, shape, materials, components, circuitry, wiring connections and contacts, as well as in the details of the illustrated circuitry, construction and method of operation may be made without departing from the spirit of the invention.

What is claimed is:

1. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a peripheral storage device coupled to the microprocessor for storing data;

a memory coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory for controlling the transfer of data from said memory; and

a parallel output port coupled to said memory and said direct memory access controller for receiving a plurality of data packets from said memory under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port further including a circuit for developing a signal for use by the external device to indicate that each data packet is available, and a circuit for receiving a signal from the external device that each data packet has been accepted.

2. The computer system of claim 1, wherein said parallel output port includes a circuit for receiving a signal from the external device that an error has occurred.

3. The computer system of claim 1, wherein said parallel output port includes a circuit for indicating to said direct memory access controller to initiate the transfer of data from said memory to said parallel output port.

4. The computer system of claim 3, wherein said direct memory access controller indicates to said parallel output port that the data transfer has been completed and where said parallel output port is coupled to said microprocessor, said parallel output port including a circuit for indicating to said microprocessor that said data transfer is completed.

5. The computer system of claim 4, wherein said parallel output port data transfer complete indicating circuit interrupts said microprocessor operation.

6. The computer system of claim 1, wherein said parallel output port is coupled to said microprocessor other than by said direct memory access controller and wherein said parallel output port includes a circuit for receiving data from said microprocessor and for providing said data to the external device.

7. The computer system of claim 6, wherein said parallel output port for receiving data from said microprocessor is disabled when said parallel output port is receiving data under control of said direct memory access controller.

8. The computer system of claim 1, wherein said parallel output port is coupled to said microprocessor and said parallel output port includes a circuit controllable by said microprocessor to initiate receipt of data by said parallel output port under control of said direct memory access controller.

9. The computer system of claim 1, wherein said direct memory access controller includes a plurality of channels and said parallel output port includes a selector circuit for selecting the channel of said direct memory access controller to be used by said parallel output port.

10. The computer system of claim 1, wherein said parallel output port can reside at a plurality of address locations.

11. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a peripheral storage device coupled to the microprocessor for storing data;

a memory means coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory means for controlling the transfer of data from said memory means;

a parallel output port means coupled to said memory means and said direct memory access controller for receiving a plurality of data packets from said memory means under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port means further including:

a means for developing a signal for use by the external device to indicate that each data packet is available;

a means for receiving a signal from the external device that each data packet has been accepted; and

a timing control circuit for controlling the operation of said parallel output port means, comprising:

a means forming a data setup time interval, wherein said data packet is provided to the external device and said signal to indicate that each data packet is available is negated, for a first predetermined time;

a means forming a data strobe time interval following said data setup time interval, wherein said data packet is provided to the external device and said signal to indicate that each data packet is available is asserted, for a second predetermined time;

a means forming an acknowledge time interval following said data strobe time interval, wherein said signal to indicate that each data packet is available is negated; and

a means forming a transfer completion time interval following said acknowledge time interval wherein said data packet is removed from provision to the external device responsive to said signal indicating that each data packet has been accepted.

12. The computer system of claim 11, wherein said parallel output port means has at least one transfer register therein.

13. The computer system of claim 12, wherein said timing control circuit furnishes the data transfer time signal to said transfer register in said parallel output port means.

14. The computer system of claim 11, wherein said external device is a parallel printer.

15. The computer system of claim 14, wherein said parallel printer has ancillary functions which are required for operation thereof and forms ancillary function signals indicative of the performance thereof.

16. The computer system of claim 15, wherein said timing control circuit includes means forming an ancillary function time interval for allowing receipt by said parallel output port means of the ancillary function signals from said parallel printer.

17. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a peripheral storage device coupled to the microprocessor for storing data;

a memory coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory for controlling the transfer of data from said memory;

a parallel output port coupled to said memory and said direct memory access controller for receiving a plurality of data packets from said memory under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port further including a circuit for developing a signal for use by the external device to indicate that each data packet is available, and a circuit for receiving a signal from the external device that each data packet has been accepted; and

a timing control circuit for controlling the operation of said parallel output port to first negate the data packet available signal, thereafter assert such a signal, then subsequently negate the data packet acceptance signal and thereafter form a transfer completion time interval signal.

18. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a peripheral device coupled to the microprocessor for performing peripheral functions;

a memory coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory for controlling the transfer of data from said memory; and

a parallel output port coupled to said memory and said direct memory access controller for receiving a plurality of data packets from said memory under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port further including a circuit for developing a signal for use by the external device to indicate that each data packet is available, and a circuit for receiving a signal from the external device that each data packet has been accepted.

19. The computer system of claim 18, wherein said parallel output port includes a circuit for receiving a signal from the external device that an error has occurred.

20. The computer system of claim 18, wherein said parallel output port includes a circuit for indicating to said direct memory access controller to initiate the transfer of data from said memory to said parallel output port.

21. The computer system of claim 18, wherein said direct memory access controller indicates to said parallel output port that the data transfer has been completed and where said parallel output port is coupled to said microprocessor, said parallel output port including a circuit for indicating to said microprocessor that said data transfer is completed.

22. The computer system of claim 21, wherein said parallel output port data transfer complete indicating circuit interrupts said microprocessor operation.

23. The computer system of claim 18, wherein said parallel output port is coupled to said microprocessor other than by said direct memory access controller and wherein said parallel output port includes a circuit for receiving data from said microprocessor and for providing said data to the external device.

24. The computer system of claim 23, wherein said parallel output port for receiving data from said microprocessor is disabled when said parallel output port is receiving data under control of said direct memory access controller.

25. The computer system of claim 18, wherein said parallel output port is coupled to said microprocessor and said parallel output port includes a circuit controllable by said microprocessor to initiate receipt of data by said parallel output port under control of said direct memory access controller.

26. The computer system of claim 18, wherein said direct memory access controller includes a plurality of channels and said parallel output port includes a selector circuit for selecting the channel of said direct memory access controller to be used by said parallel output port.

27. The computer system of claim 18, wherein said parallel output port can reside at a plurality of address locations.

28. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a peripheral device coupled to the microprocessor for performing peripheral functions;

a memory means coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory means for controlling the transfer of data from said memory means;

a parallel output port means coupled to said memory means and said direct memory access controller for receiving a plurality of data packets from said memory means under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port means further including:

a means for developing a signal for use by the external device to indicate that each data packet is available;

a means for receiving a signal from the external device that each data packet has been accepted; and

a timing control circuit for controlling the operation of said parallel output port means, comprising:

a means forming a data setup time interval, wherein said data packet is provided to the external device and said signal to indicate that each data packet is available is negated, for a first predetermined time;

a means forming a data strobe time interval following said data setup time interval, wherein said data packet is provided to the external device and said signal to indicate that each data packet is available is asserted, for a second predetermined time;

a means forming an acknowledge time interval following said data strobe time interval, wherein said signal to indicate that each data packet is available is negated; and

a means forming a transfer completion time interval following said acknowledge time interval wherein said data packet is removed from provision to the external device responsive to said signal indicating that each data packet has been accepted.

29. The computer system of claim 28, wherein said parallel output port means has at least one transfer register therein.

30. The computer system of claim 29, wherein said timing control circuit furnishes the data transfer time signal to said transfer register in said parallel output port means.

31. The computer system of claim 21, wherein said external device is a parallel printer.

32. The computer system of claim 31, wherein said parallel printer has ancillary functions which are required for operation thereof and forms ancillary function signals indicative of the performance thereof.

33. The computer system of claim 32, wherein said timing control circuit includes means forming an ancillary function time interval for allowing receipt by said parallel output port means of the ancillary function signals from said parallel printer.

34. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a peripheral device coupled to the microprocessor for performing peripheral functions;

a memory coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory for controlling the transfer of data from said memory;

a parallel output port coupled to said memory and said direct memory access controller for receiving a plurality of data packets from said memory under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port further including a circuit for developing a signal for use by the external device to indicate that each data packet is available, and a circuit for receiving a signal from the external device that each data packet has been accepted; and

a timing control circuit for controlling the operation of said parallel output port to first negate the data packet available signal, thereafter assert such a signal, then subsequently negate the data packet acceptance signal and thereafter form a transfer completion time interval signal.

35. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a video system coupled to the microprocessor for displaying data;

a memory coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory for controlling the transfer of data from said memory; and

a parallel output port coupled to said memory and said direct memory access controller for receiving a plurality of data packets from said memory under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port further including a circuit for developing a signal for use by the external device to indicate that each data packet is available, and a circuit for receiving a signal from the external device that each data packet has been accepted.

36. The computer system of claim 35, wherein said parallel output port includes a circuit for receiving a signal from the external device that an error has occurred.

37. The computer system of claim 35, wherein said parallel output port includes a circuit for indicating to said direct memory access controller to initiate the transfer of data from said memory to said parallel output port.

38. The computer system of claim 35, wherein said direct memory access controller indicates to said parallel output port that the data transfer has been completed and where said parallel output port is coupled to said microprocessor, said parallel output port including a circuit for indicating to said microprocessor that said data transfer is completed.

39. The computer system of claim 38, wherein said parallel output port data transfer complete indicating circuit interrupts said microprocessor operation.

40. The computer system of claim 35, wherein said parallel output port is coupled to said microprocessor other than by said direct memory access controller and wherein said parallel output port includes a circuit for receiving data from said microprocessor and for providing said data to the external device.

41. The computer system of claim 40, wherein said parallel output port for receiving data from said microprocessor is disabled when said parallel output port is receiving data under control of said direct memory access controller.

42. The computer system of claim 35, wherein said parallel output port is coupled to said microprocessor and said parallel output port includes a circuit controllable by said microprocessor to initiate receipt of data by said parallel output port under control of said direct memory access controller.

43. The computer system of claim 35, wherein said direct memory access controller includes a plurality of channels and said parallel output port includes a selector circuit for selecting the channel of said direct memory access controller to be used by said parallel output port.

44. The computer system of claim 35, wherein said parallel output port can reside at a plurality of address locations.

45. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a video system coupled to the microprocessor for displaying data;

a memory means coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory means for controlling the transfer of data from said memory means;

a parallel output port means coupled to said memory means and said direct memory access controller for receiving a plurality of data packets from said memory means under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port means further including:

a means for developing a signal for use by the external device to indicate that each data packet is available;

a means for receiving a signal from the external device that each data packet has been accepted; and

a timing control circuit for controlling the operation of said parallel output port means, comprising:

a means forming a data setup time interval, wherein said data packet is provided to the external device and said signal to indicate that each data packet is available is negated, for a first predetermined time;

a means forming a data strobe time interval following said data setup time interval, wherein said data packet is provided to the external device and said signal to indicate that each data packet is available is asserted, for a second predetermined time;

a means forming an acknowledge time interval following said data strobe time interval, wherein said signal to indicate that each data packet is available is negated; and

a means forming a transfer completion time interval following said acknowledge time interval wherein said data packet is removed from provision to the external device responsive to said signal indicating that each data packet has been accepted.

46. The computer system of claim 45, wherein said parallel output port means has at least one transfer register therein.

47. The computer system of claim 46, wherein said timing control circuit furnishes the data transfer time signal to said transfer register in said parallel output port means.

48. The computer system of claim 45, wherein said external device is a parallel printer.

49. The computer system of claim 48, wherein said parallel printer has ancillary functions which are required for operation thereof and forms ancillary function signals indicative of the performance thereof.

50. The computer system of claim 49, wherein said timing control circuit includes means forming an ancillary function time interval for allowing receipt by said parallel output port means of the ancillary function signals from said parallel printer.

51. A computer system for communicating with an external device in a parallel format, comprising:

a microprocessor;

a video system coupled to the microprocessor for displaying data;

a memory coupled to said microprocessor for storing instructions and data for said microprocessor and data to be communicated to the external device;

a direct memory access controller coupled to said memory for controlling the transfer of data from said memory;

a parallel output port coupled to said memory and said direct memory access controller for receiving a plurality of data packets from said memory under control of said direct memory access controller and for providing said data packets to the external device, said parallel output port further including a circuit for developing a signal for use by the external device to indicate that each data packet is available, and a circuit for receiving a signal from the external device that each data packet has been accepted; and

a timing control circuit for controlling the operation of said parallel output port to first negate the data packet available signal, thereafter assert such a signal, then subsequently negate the data packet acceptance signal and thereafter form a transfer completion time interval signal.

* * * * *

EXHIBIT B

US006233691B1

(12) **United States Patent**
Atkinson

(10) **Patent No.: US 6,233,691 B1**
(45) **Date of Patent: May 15, 2001**

(54) **APPARATUS FOR REDUCING COMPUTER SYSTEM POWER CONSUMPTION**

(75) Inventor: **Lee Warren Atkinson**, Houston, TX (US)

(73) Assignee: **Compaq Computer Corporation**, Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/480,101**

(22) Filed: **Jan. 10, 2000**

Related U.S. Application Data

(63) Continuation of application No. 08/166,609, filed on Dec. 13, 1993, now Pat. No. 6,029,249, which is a continuation of application No. 07/809,301, filed on Dec. 17, 1991, now abandoned.

(51) **Int. Cl.**[7] **G06F 1/32**
(52) **U.S. Cl.** **713/323**; 713/322
(58) **Field of Search** 713/300, 320, 713/321–323, 501, 600; 709/100; 710/1; 340/636; 307/66

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,317,181		2/1982	Teza et al.	364/707
4,670,837		6/1987	Sheets	364/200
4,698,748		10/1987	Juzswik et al.	364/200
4,819,164		4/1989	Branson	364/200
4,980,836		12/1990	Carter et al.	364/483
5,142,684	*	8/1992	Perry et al. .	
5,167,024		11/1992	Smith et al.	395/375
5,487,181	*	1/1996	Dailey et al. .	
5,655,127	*	8/1997	Rabe et al. .	
6,029,249	*	2/2000	Atkinson .	

FOREIGN PATENT DOCUMENTS

0451661	10/1991	(EP) .
2151950	6/1990	(JP) .

OTHER PUBLICATIONS

Intel Corp., *386 SL Microprocessor Superset, System Design Guide*, Chapter 14 "System and Power Management," 14–1 to 14–23, 1990.

Intel Corp., *396 SL Microprocessor Superset, Programmers Reference Manual*, Chapter 6 "Systems and Power Management," pp. 6–1 to 6–50; 10–4 to 10–7, 1990.

Technique for Monitoring a Computer System's Activity for the Purpose of Power Management of a DOS–Compatible System, IBM Technical Disclosure Bulletin, vol. 33, No. 4, Sep. 1990, pp. 474–477.

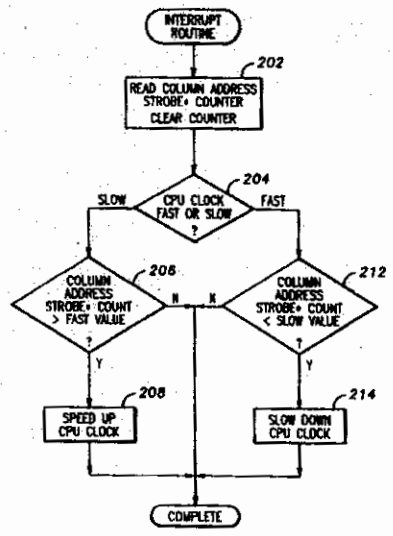
* cited by examiner

Primary Examiner—Gopal C. Ray
(74) *Attorney, Agent, or Firm*—Akin, Gump, Strauss, Hauer & Feld, LLP

(57) **ABSTRACT**

A battery powered computer system determines when the system is not in use by monitoring various events associated with the operation of the system. The system preferably monitors the number of cache read misses and write operations, i.e., the cache hit rate, and reduces the system clock frequency when the cache hit rate rises above a certain level. When the cache hit rate is above a certain level, then it can be assumed that the processor is executing a tight loop, such as when the processor is waiting for a key to be pressed and then the frequency can be reduced without affecting system performance. Alternatively, the apparatus monitors the occurrence of memory page misses, I/O write cycles or other events to determine the level of activity of the computer system.

23 Claims, 4 Drawing Sheets



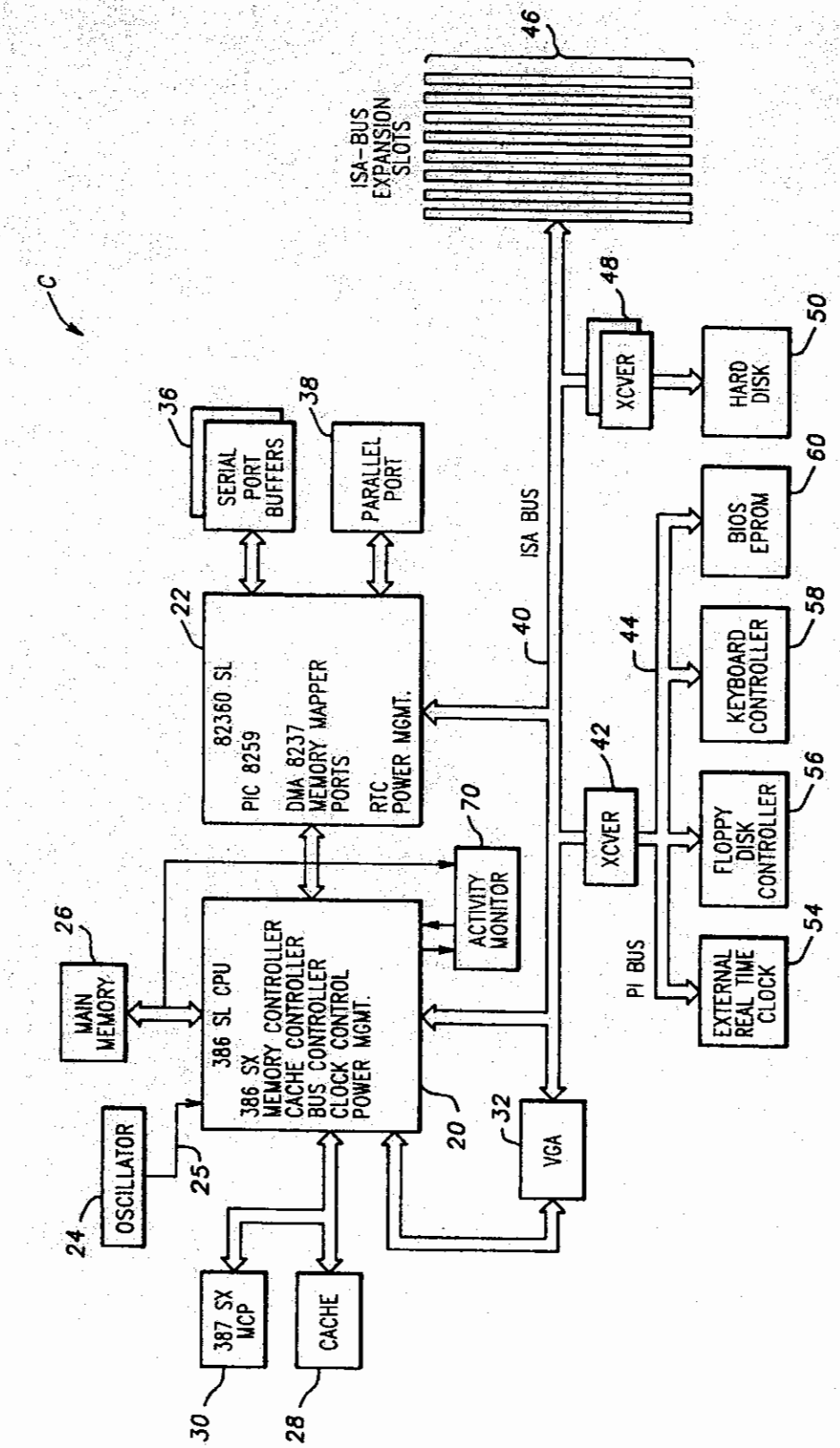

C
ISA-BUS EXPANSION SLOTS
46
ISA BUS
40
XCVER
48
HARD DISK
50
SERIAL PORT BUFFERS
36
PARALLEL PORT
38
BIOS EPROM
60
82360 SL
PIC 8259
DMA 8237
MEMORY MAPPER PORTS
RTC
POWER MGMT.
22
KEYBOARD CONTROLLER
58
44
XCVER
42
FLOPPY DISK CONTROLLER
56
ACTIVITY MONITOR
70
PI BUS
EXTERNAL REAL TIME CLOCK
54
MAIN MEMORY
26
386 SL CPU
386 SX
MEMORY CONTROLLER
CACHE CONTROLLER
BUS CONTROLLER
CLOCK CONTROL
POWER MGMT.
20
VGA
32
OSCILLATOR
25
24
387 SX MCP
30
CACHE
28

FIG. 1

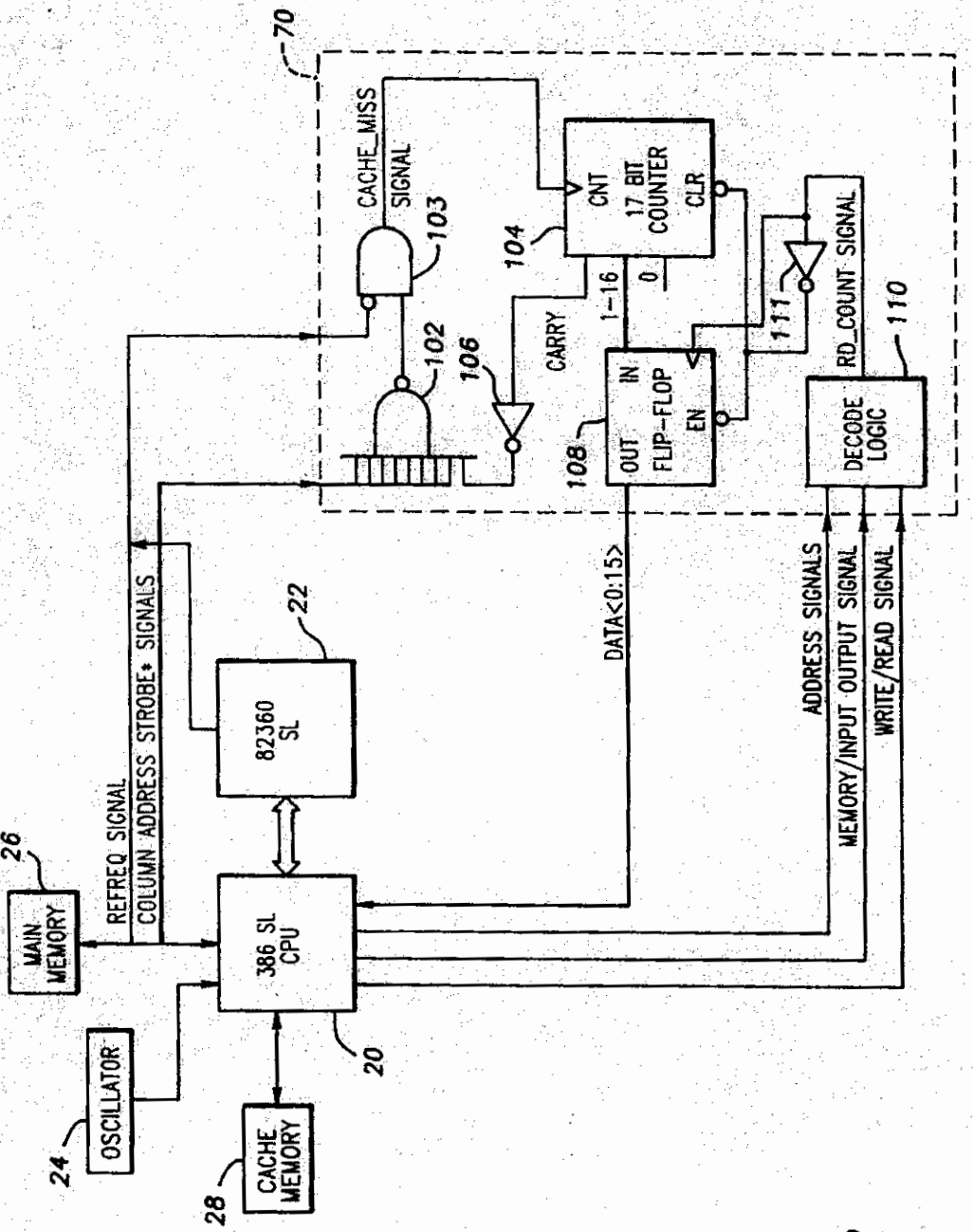

70
CACHE_MISS SIGNAL
103
104
CNT
17 BIT COUNTER
CLR
1–16
0
CARRY
102
106
108
IN
OUT
FLIP–FLOP
EN
111
RD_COUNT SIGNAL
110
DECODE LOGIC
DATA<0:15>
ADDRESS SIGNALS
MEMORY/INPUT OUTPUT SIGNAL
WRITE/READ SIGNAL
22
82360 SL
REFREQ SIGNAL
COLUMN ADDRESS STROBE* SIGNALS
26
MAIN MEMORY
386 SL CPU
20
OSCILLATOR
24
CACHE MEMORY
28

FIG. 2

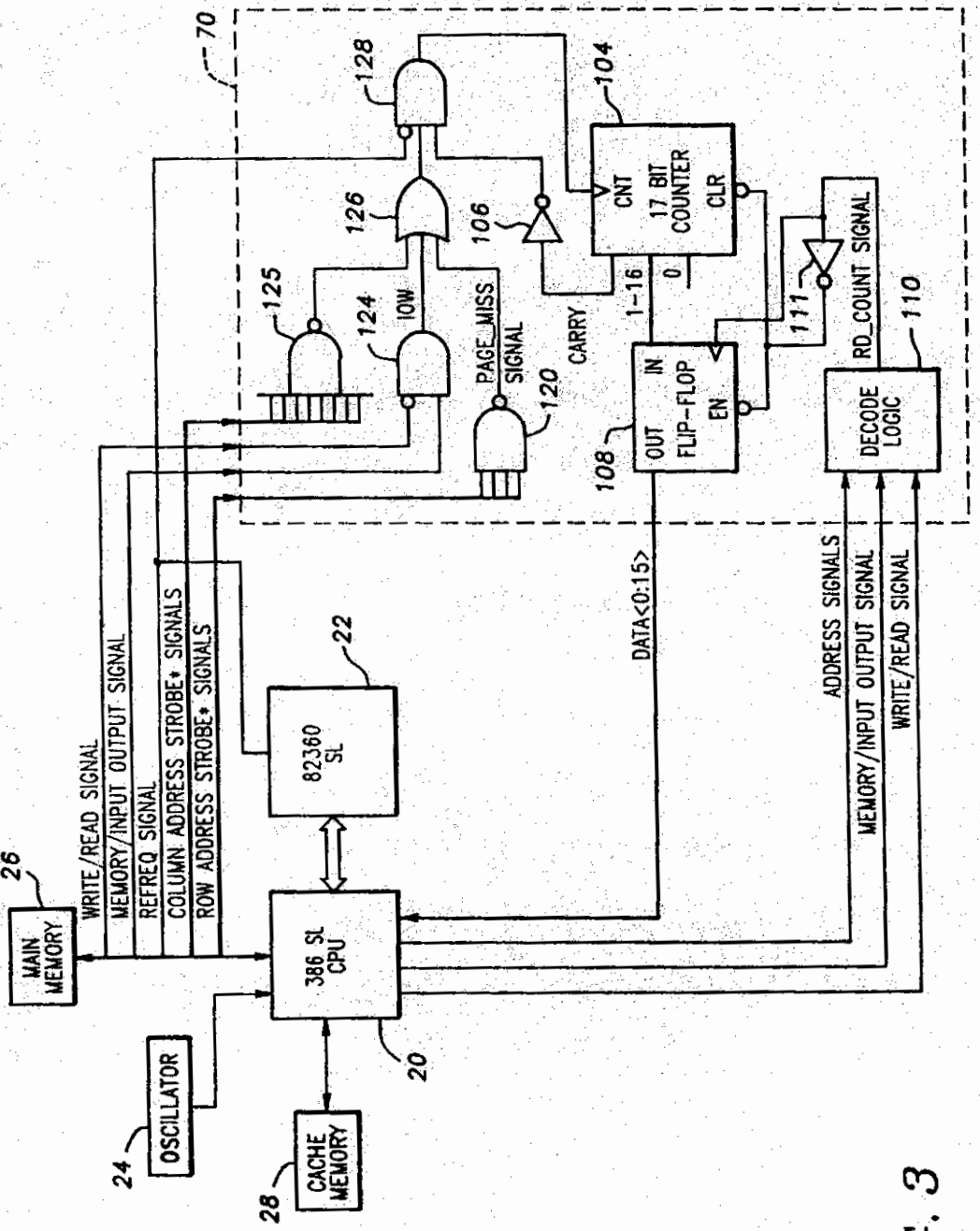

70
128
104
126
106
125
124
IOW
PAGE_MISS SIGNAL
120
CARRY
CNT
17 BIT COUNTER
CLR
1-16
0
111
RD_COUNT SIGNAL
110
IN
OUT
FLIP-FLOP
EN
108
DECODE LOGIC
DATA<0:15>
ADDRESS SIGNALS
MEMORY/INPUT OUTPUT SIGNAL
WRITE/READ SIGNAL
22
82360 SL
26
MAIN MEMORY
WRITE/READ SIGNAL
MEMORY/INPUT OUTPUT SIGNAL
REFREQ SIGNAL
COLUMN ADDRESS STROBE* SIGNALS
ROW ADDRESS STROBE* SIGNALS
386 SL CPU
20
OSCILLATOR
24
CACHE MEMORY
28

FIG. 3

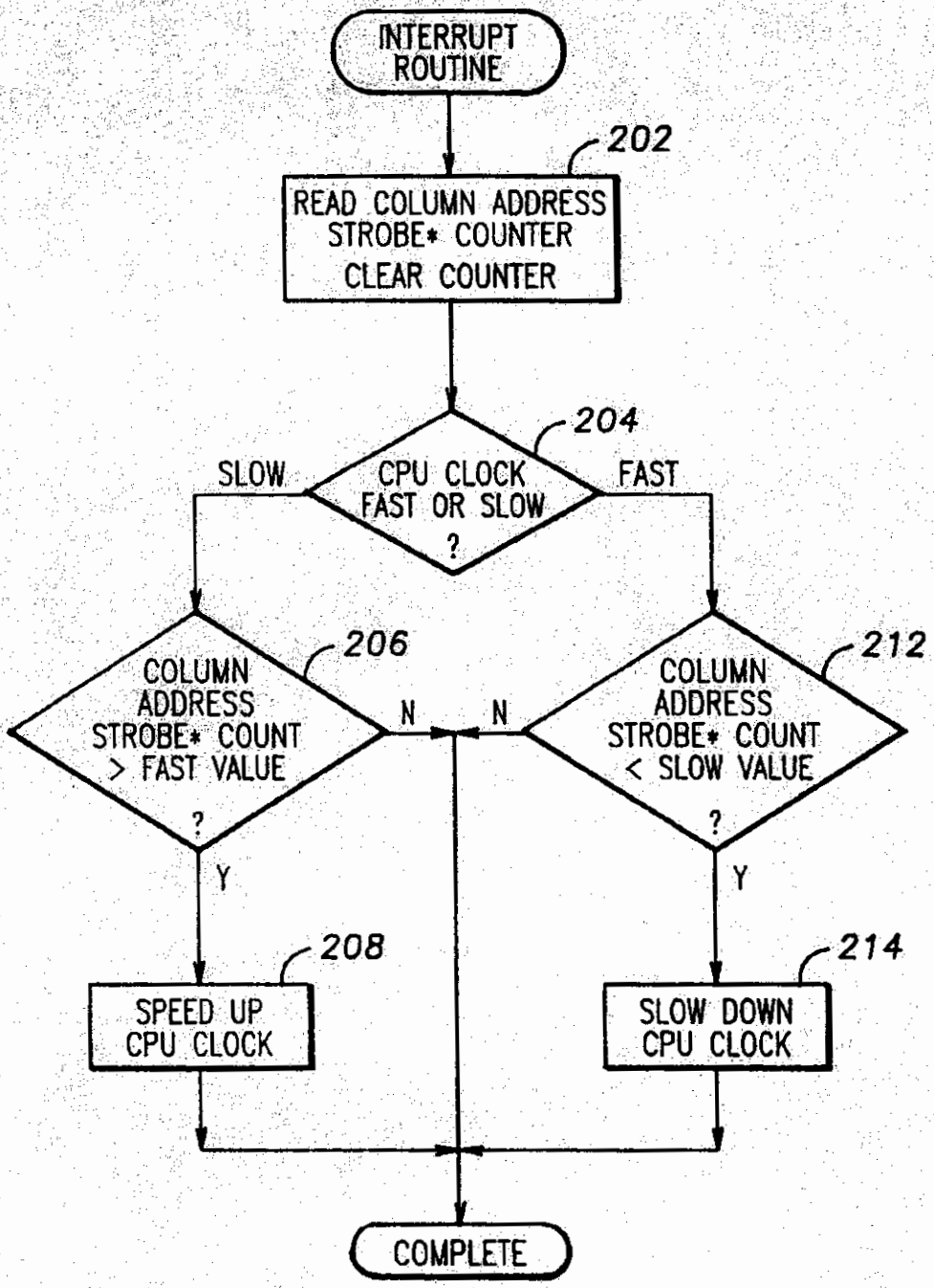

INTERRUPT ROUTINE
202
READ COLUMN ADDRESS STROBE* COUNTER
CLEAR COUNTER
204
SLOW
CPU CLOCK FAST OR SLOW ?
FAST
206
COLUMN ADDRESS STROBE* COUNT > FAST VALUE ?
N
N
212
COLUMN ADDRESS STROBE* COUNT < SLOW VALUE ?
Y
Y
208
SPEED UP CPU CLOCK
214
SLOW DOWN CPU CLOCK
COMPLETE

FIG. 4

APPARATUS FOR REDUCING COMPUTER SYSTEM POWER CONSUMPTION

RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 08/166,609, filed Dec. 13, 1993, U.S. Pat. No. 6,029,249, which is a continuation of U.S. application Ser. No. 07/809,301, filed Dec. 17, 1991, abandoned.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to battery powered computer systems, and more particularly, to circuits and methods for reducing the power consumption of the computer system.

2. Description of the Prior Art

Portable computer systems are rapidly developing the capabilities of conventional desktop or floor mounted personal computer systems. Hard disk units are being integrated into portable computers because of the large amounts of information being processed and the large size of many application programs. A floppy disk unit is integrated in the vast majority of portable computers, even if a hard disk unit is installed, to allow loading of information and use of applications requiring key disks, and also to allow use of diagnostic programs. Modems have been integrated into portable computers for some time to allow communications and information transfer between the user and a remote location, for example, the home office. The displays in portable computer systems are becoming much more elaborate and readable. The pixel count on the standard liquid crystal displays (LCD's) utilized is increasing, as is the viewing angle. The use of backlighting allows use of LCD's in low light environments and improves the contrast ratio of the display. More complex circuitry is being installed in portable computers to support these improved peripheral devices and to support the increased speeds and capabilities of the microprocessors utilized in portable computer systems.

The various peripheral devices and high speed circuitry mentioned above consume large amounts of power when operating. This has resulted in problems in portable computer systems because these systems are generally desired to be used in locations where alternating current is not available. This has made it very difficult to provide all the possible functionality available and yet have an acceptable battery life when the portable computer system is battery powered. Using CMOS components helped reduce the power consumption of the circuitry, but even the use of CMOS components is insufficient at the clock speeds and performance levels of available circuitry. Therefore a dilemma arises whether to provide lesser functionality with longer battery life or greater functionality with lesser battery life or even no battery operation.

Various alternatives were tried to resolve the problem. For example, the International Business Machines (IBM) Corporation PC Convertible included a switch which the user could press to place the computer system in a standby mode. However, the PC convertible was relatively simple, with a low level of functionality as compared to what is currently available, and the requirement of a user action to reduce power consumption limited its use to circumstances where the user remembered to depress the switch. Blanking the display after a period of keyboard inactivity saved power as well as prolonged the life of the display and was widely utilized. A hard disk unit was developed which reduced the power used by the controlling electronics by utilizing only certain portions of the track for servo information and turning off the read channel circuitry until just before a servo burst was expected. Additionally, a programmable value could be provided to the hard disk unit so that after a given inactivity interval defined by this value, the hard disk unit was allowed to spin down and all but some interface circuitry was shut down. While these alternatives did provide some relief, they were not complete solutions to satisfactorily resolve the dilemma, and design tradeoffs still were forced to occur.

U.S. Pat. No. 4,980,836 to Carter et al. discloses an apparatus for reducing power consumption in computer systems. The apparatus monitors the address bus to determine when selected peripheral devices have not been accessed for a preset amount of time. When the preset amount of time has passed, the system powers itself down and disables the system clock, placing the system in a standby mode. The system clock could be stopped in this invention because the preferred embodiment of this invention used a static CMOS processor and chip set. If there was sufficient energy in the batteries, the system could be awakened by the user depressing a standby switch. Computer systems which do not use a static CMOS processor or chip set generally reduce the clock frequency when a preset amount of time of address bus inactivity has passed. Reducing the clock frequency during inactive periods reduces power consumption during this time. However, the frequency of peripheral device accesses is not a completely reliable indicator of inactivity of a computer system. Thus, in some instances the system clocking signal may be reduced in frequency or disabled during a period of high computer system activity. Therefore, a method is needed whereby other elements or events of the computer system can be monitored to more reliably determine the activity level of the computer system so that the system clock can properly be adjusted to reduce power consumption.

SUMMARY OF THE INVENTION

A battery powered computer system according to the present invention determines when the system is not in use by monitoring various events associated with the operation of the computer system. In the preferred embodiment, the system monitors the number of cache read misses and write operations, i.e., the cache hit rate, and reduces the system clock frequency when the cache hit rate rises above a certain level. When the cache read hit rate is above a certain level, then it can be assumed that the processor is executing a tight loop, such as when the processor is waiting for a key to be pressed. In an alternate embodiment of the invention, the system monitors other events in addition to, or instead of, the cache read hit rate, such as the occurrence of page hits or input/output (I/O) write cycles, to determine the level of activity of the computer system.

The system according to the preferred embodiment includes a frequency switching circuit, an event counter, and a periodic timer. The event counter is preferably used to measure the incidence of cache read misses and write operations and may also optionally be used to count the number of page misses and memory or I/O writes as desired. The event counter includes an overflow or carry line which prevents any further incrementing of the counter once the maximum number of counts is reached to prevent the counter from overflowing. The periodic timer instructs the CPU via a system interrupt to periodically monitor and compare the contents of the event counter. Every event increments the counter and, the more events, the more

processor activity that is presumed. When the periodic timer issues a system interrupt, the CPU reads the contents of the counter and compares the event activity with a predetermined value. If the number of events is higher than the predetermined value, then the processor switches the operating frequency of the system to a high frequency if the system is not already operating at this high frequency. A lower event count causes the frequency switching circuit to switch to a lower frequency to conserve power if the system is not already operating at this low frequency.

The invention allows the battery powered operating period of a computer system to be greatly extended without requiring any input from the user and without any noticeable loss in processing power. This allows a battery powered computer system to have advanced capabilities and functionality while still having a satisfactory battery operating interval.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the invention can be obtained when the following detailed description of the preferred embodiment is considered in conjunction with the following drawing in which:

FIG. 1 is a schematic block diagram of a computer system incorporating the present invention;

FIG. 2 is a more detailed schematic diagram of a portion of the computer system of FIG. 1;

FIG. 3 is a schematic diagram of a portion of the computer system of FIG. 1 according to an alternate embodiment of the invention; and

FIG. 4 is a flowchart diagram of a sequence for controlling the operation of the computer system of FIG. 1 according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to FIG. 1, a computer system C according to the preferred embodiment of the invention is shown. The computer system C is preferably based on the 386 SL chip set produced by Intel Corporation (Intel). The 386 SL chip set comprises two chips, a 386 SL CPU chip (CPU chip) 20 and a 82360 SL chip 22. The CPU chip 20 includes an 80386 SX microprocessor, a memory controller, a cache controller, a bus controller, clock control circuitry, and power management circuitry. The 82360 SL chip 22 includes a programmable interrupt controller (PIC), direct memory access (DMA) controller, a memory mapper, various ports, a real time clock (RTC) and power management circuitry. For more information on the 386 SL chip set, please see the Intel 386 SL Microprocessor Superset System Design Guide 1990 edition; the 386 SL Microprocessor Superset Programmers Reference Manual, 1990 edition; and the 386 SL Microprocessor Superset Data Book; all published by Intel.

An oscillator 24 is connected to the CPU chip 20. The oscillator preferably operates at 20 Megahertz (MHz) and provides a 20 MHz clocking signal 25 to the CPU chip 20. Main memory 26, cache memory 28 and a math coprocessor (MCP) 30 are also preferably coupled to the CPU chip 20. The math coprocessor 30 is preferably a 387 SX coprocessor produced by Intel. The cache memory is preferably operated as a write-through cache memory. A video graphics array (VGA) controller 32 is coupled to the CPU chip 20. Serial port buffers 36 and a parallel port 38 are coupled to the 82360 SL chip 22.

The CPU chip 20 and the 82360 SL chip 22 are each connected to an I/O bus 40 based on the industry standard architecture (ISA). However, other bus architectures are also contemplated. The ISA bus 40 is connected through a transceiver 42 to a peripheral interface (PI) bus 44. The VGA chip 32 is connected to the ISA bus 40. The ISA bus 40 may include a plurality of ISA bus expansion slots 46 if the present invention is used in a desktop computer system. The slots 46 are generally omitted in portable computers. The ISA bus 40 is connected through a transceiver 48 to a hard disk 50. Various logic is coupled to the PI bus 44, including an external real time clock 54, a floppy disk controller 56, a keyboard controller 58 and BIOS EPROM 60. The floppy disk controller 56 and keyboard controller 58 control operation of a floppy disk unit and keyboard, respectively (both not shown). The BIOS EPROM holds the basis input/output (I/O) system software as well as system-specific initialization and configuration software. This is an exemplary computer system and other designs and architectures could be utilized.

The CPU chip 20 includes an active high input referred to as the TURBO input (not shown). When the TURBO input receives a logic high signal, the CPU chip 20 enters "turbo mode," and the microprocessor executes at a clock speed defined by a bit field in a register in the CPU chip 20 referred to as the CPUWRMODE register (not shown). When the TURBO input receives a logic low signal, the CPU chip 20 enters "de-turbo mode" and executes at a reduced clock speed of ½ or ¼ as defined by a bit in a register. The CPUWRMODE register operates in conjunction with the power management circuitry inside the CPU chip 20 to control the frequency of the clocking signal 25 provided to the microprocessor. In the preferred embodiment, the turbo input receives a logic high value, and thus the CPUWRMODE register determines the frequency of the clocking signal 25 provided to the microprocessor.

Bits 5 and 4 of the CPUWRMODE register determine the speed with which the microprocessor clock operates. When bits 5 and 4 of the CPUWRMODE register are each set to 0, then the clock speed is the speed of the signal received by the oscillator 24, preferably 20 MHz. When bits 5 and 4 of the CPUWRMODE register are set to 0 and 1 respectively, the clock speed provided to the microprocessor is one half of the frequency of the signal received from the oscillator 24, i.e., 10 MHz. When bits 5 and 4 of the CPUWRMODE register are 1 and 0 respectively, the clock speed provided to the microprocessor is one fourth of the frequency of the signal received from the oscillator 24, i.e., 5 MHz. When bits 5 and 4 of the CPUWRMODE register each have a logic 1 value, then the clock speed provided to the CPU is one eighth the frequency of the signal received from the oscillator 24, i.e. 2.5 MHz.

The computer system C also includes activity monitor logic 70 according to the present invention which is coupled to the CPU chip 20. The activity monitor logic 70 receives signals from the CPU chip 20 as well as signals provided from the CPU chip 20 to the main memory 26. The activity monitor logic 70 monitors events associated with the microprocessor to determine the activity level of the system. In the preferred embodiment, the activity monitor logic 70 monitors the number of cache read misses and write operations during preset periods of time, and the CPU chip 20 reduces the system clock frequency when the cache hit rate rises above a certain level. When the cache read hit rate is above a certain level, then it can be assumed that the processor is executing a tight loop, such as when the processor is waiting for a key to be pressed. In this instance, the system clock frequency can be reduced to reduce power consumption without affecting system performance. In an alternate

embodiment of the invention, the activity monitor logic **70** monitors other events such as page misses or memory and I/O write operations, among others. It is contemplated that the various types of events may be counted either individually or in combination.

Referring now to FIG. **2**, the activity monitor logic **70** according to the preferred embodiment of the invention is shown. In the description that follows a signal name followed by an asterisk indicates that the signal is asserted when it has a logic low level. The activity monitor logic **70** receives column address strobe (CAS*) signals provided from the CPU chip **20** to the main memory **26**. The CAS* signals are essentially memory select signals that are used in paged memory systems. When a cache read hit occurs, the requested data can be provided directly from the cache memory **28** to the microprocessor, and no CAS* signals are asserted by the CPU chip **20**. Also, since the cache memory is operated as a write-through cache, each write operation, whether it be a write hit or write miss, requires a memory cycle to the main memory **26**. Therefore, the CAS* signals are asserted on all write operations in the preferred embodiment. Thus, the CAS* signals are used to determine when cache read miss operations and write operations occur, i.e., when main memory cycles occur.

In the preferred embodiment, the CPU chip **20** generates eight CAS* signals. The CAS* signals are provided to eight inputs of a nine input NAND gate **102**, in effect ORing these signals together. The output of the NAND gate **102** is provided to an input of a two input AND gate **103**. A signal referred to as REFREQ output from the **82360** SL chip **22** is preferably provided to the other input of the AND gate **103**, which is an inverted input. The REFREQ signal indicates, when asserted high, that a memory refresh operation is occurring. The inverted REFREQ signal is ANDed with the output of the NAND gate **102** to prevent memory refresh operations from being counted as cache read miss or write operations.

The output of the AND gate **103** is a signal referred to as CACHE_MISS, which is provided to the clock input of a **17** bit counter **104** referred to as the CAS* counter. The CACHE_MISS signal is therefore asserted when a main memory cycle occurs, which is whenever a cache read miss or a write operation occurs. The CAS* counter **104** is incremented each time the CACHE_MISS signal is asserted. The CAS* counter **104** outputs a signal referred to as CARRY which is provided through an inverter **106** to the seventh input of the NAND gate **102**. The CARRY signal is asserted when the maximum number of counts is reached by the counter **104**, which is 2^{17}. When the CARRY signal is asserted, the CACHE_MISS signal is prevented from changing state and thus further counting by the counter **104** is disabled. The CAS* counter **104** provides data signals **1–16** to the inputs of a **16** bit flip-flop **108**. Data signal **0** output from the CAS* counter is preferably left unconnected. The outputs of the flip-flop **108** are coupled to data bus signals referred to as DATA<0:15>, which are coupled to the CPU chip **20**.

Address signals, a write/read (W/R) signal and a memory/input output (M/IO) signal are output from the CPU chip **20** and provided to decode logic **110** in the activity monitor logic **70**. The decode logic **110** outputs a signal referred to as RD_COUNT which is provided to a clock input of the flip-flop **108**. The RD_COUNT signal is also provided through an inverter **111** to an inverted clear input of the counter **104** and to an inverted enable input of the flip-flop **108**. Therefore, when the CPU chip **20** desires to read the counter **104**, it outputs the appropriate address, the W/R signal and the M/IO signal to the decode logic **110**, which then asserts the RD_COUNT signal to the counter **104** and the flip-flop **108**. The asserted RD_COUNT signal enables the flip-flop **108** to latch in data from the counter **104** and transmit the data to the CPU chip **20** and simultaneously operates to clear the contents of the counter **104**.

In an alternate embodiment, memory page misses and/or I/O write operations may also be counted. Memory page miss operations with the preferred microprocessor are determined by determining if any of the **4** row address strobe or RAS* signals are asserted low during non-refresh cycles. The asserted RAS* signals are indicative of page miss operations. As shown in FIG. **3**, the RAS* signals are provided from the CPU chip **20** to the inputs of a four input NAND gate **120**. The output of the NAND gate **120** is the page miss indication, which is referred to as the PAGE_MISS signal. The I/O write indication is accomplished in the following manner. The W/R signal is connected to an input of a two input AND gate **124**. The other input of the AND gate **124** is inverted and receives the M/IO signal. The output of the AND gate **124** generates a signal referred to as IOW which indicates, when asserted high, that an I/O write operation is occurring.

The PAGE_MISS and IOW signals are connected to inputs of a three input OR gate **126**. The eight CAS* signals are connected to the input of an eight input NAND gate **125** whose output is connected to an input of the OR gate **126**. The output of the OR gate **126** is connected to an input of a three input AND gate **128**. A second input of the AND gate **128** receives the inverted CARRY signal. The third inverted input of the AND gate **128** receives the REFREQ signal. The output of the AND gate **128** is provided to the clock input of the counter **104**. Therefore, during memory refresh operations or when the CARRY signal is asserted, the output of the AND gate **128** is negated low, and thus the counter **104** does not count at these times.

In this embodiment, cache read miss and write operations, page miss operations, and I/O write operations are all monitored as events. It is noted that any one of these events can be monitored either individually or in combination. In yet another alternate embodiment, each signal can be provided to separate counters arranged like the counter **104**, and the outputs of the separate counters can be used to determine system activity.

The **82360** SL chip **22** includes an interrupt timer (not shown) which is used to generate time of day interrupts to the microprocessor approximately 18.2 times per second. The interrupt routine invoked by the timer is also used according to the preferred embodiment of the invention to periodically adjust the frequency of the clocking signal **25** received by the microprocessor, as is explained below.

Referring now to FIG. **4**, a portion of the software routine that is executed when the time of day interrupt routine is invoked is shown. It is understood that the time of day interrupt routine may perform other operations than those shown. When the time of day interrupt signal is asserted, the CPU chip **20** reads the CAS* counter **104** in step **202**. The counter **104** is also cleared in step **202**. In an alternative embodiment, the separate counters providing the CAS* count, the memory page miss count, and the I/O write count are read in step **202**. In step **204**, the microprocessor reads bits **5** and **4** of the CPUWRMODE register to determine if the clock signal provided to the CPU chip **20** is operating at a fast or slow frequency. In the preferred embodiment, the microprocessor only writes either values 0,0 or 1,0 to bits **5** and **4**, respectively, of the CPUWRMODE register, thus

providing the clocking signal **25** at either full frequency or one quarter frequency to the microprocessor. The CPUWR-MODE register is programmed depending on the cache read hit rate as determined by the CAS* counter **104** in relation to certain comparison values. Thus, in this embodiment, the power management logic toggles between a full clock speed or fast speed, preferably 20 MHz, and a one quarter frequency clock speed or slow speed, preferably 5 MHz. However, it is noted that all four clock speed frequencies may be utilized.

If the clocking signal **25** provided to the CPU is determined to be operating at the slow speed in step **204**, then in step **206** the microprocessor determines if the CAS* count value received from the CAS* counter **104** is greater than a fast comparison value stored inside the CPU chip **20**. Alternatively, a function based on the three separate count values could be evaluated. If the CAS* count value exceeds the fast comparison value in step **206**, then in step **208** the power management logic in the CPU chip **20** speeds up the clock signal **25** provided to the microprocessor to the full clock speed, and the routine completes. If the CAS* count value is not greater than the fast comparison value in step **206**, then the interrupt routine completes.

If the clocking signal **25** provided to the microprocessor is determined to be operating at the fast speed in step **204**, then control proceeds to step **212** where the microprocessor determines whether the CAS* count value is less than a slow comparison value stored in the CPU chip **20**. If the CAS* count value is less than the slow comparison value in step **212**, then the power management logic in the CPU chip **20** slows down the clocking signal **25** provided to the microprocessor in step **214** to one quarter speed, and the routine completes. If the CAS* count value is not less than the slow comparison value in step **212**, then the routine completes.

The fast and slow comparison values are preferably programmable inside the CPU chip **20**. In the preferred embodiment, the fast and slow comparison values are the same value. In an alternate embodiment, the slow comparison value is proportionally less that the fast value, i.e., one fourth less to reflect a true hit/miss ratio. In addition, since the comparison values can be user-adjusted inside the CPU chip **20**, the values could be derived from characterization of common user applications.

As described above, the preferred embodiment utilizes the CPU chip **20** in "turbo mode" and uses software to perform the activity comparisons and speed changes. In an alternative embodiment, additional, registers can be used to contain the comparison values and an additional timer can be used to define the comparison interval. Hardware comparators are configured to perform the comparison logic described above at the appropriate time and change the state of the TURBO input based on the comparison to speed up or slow down the CPU chip **20** as appropriate.

Thus, the computer system according to the present invention monitors various power consumption related events, indicates certain changes to the user and enters an inactivity state upon an appropriate period of time after monitored system devices have been used.

The foregoing disclosure and description of the invention are illustrative and explanatory thereof, and various changes in the components, circuit elements, or flowcharts may be made without departing from the spirit of the invention.

I claim:

1. A computer system capable of operation in different power consumption modes, the computer system operating in a normal power consumption mode when the computer system is subject to a normal level of activity and operating in a reduced power consumption mode when the computer system is subject to a reduced level of activity, comprising:

a processor;

a mass storage device coupled to the processor; and

a power management circuit coupled to the processor, the power management circuit adapted to reduce the power consumption of the computer system responsive to a rate of processor events being below a predetermined low event threshold and adapted to increase power consumption of the computer system responsive to the rate of processor events and being above a predetermined high event threshold, the power management circuit determining the rate of processor events based on processor signals.

2. The computer system of claim **1**, wherein the processor operates at a system clock frequency, and wherein the power management circuit reduces power consumption by reducing the system clock frequency.

3. The computer system of claim **2**, wherein the power management circuit increases power consumption by increasing the system clock frequency.

4. The computer system of claim **3**, wherein the system clock frequency is adjustable to operate at a plurality of different system clock frequencies.

5. The computer system of claim **1**, further comprising:

a counter timer for determining the rate of processor events.

6. The computer system of claim **1**, wherein the low event threshold and the high event threshold are the same value.

7. The computer system of claim **1**, further comprising:

cache memory coupled to the processor; and

wherein the processor events comprise cache miss operations and memory write operations.

8. The computer system of claim **1**, further comprising:

cache memory coupled to the processor; and

input/output devices coupled to the processor

wherein the processor events are selected from the group consisting of cache miss operations and memory write operations, memory page miss operations and input/output write operations.

9. The computer system of claim **8**, further comprising:

decode logic coupled to the processor to select the type of processor event from which the rate of processor events is determined.

10. The computer system of claim **1**, wherein the computer system is a portable computer.

11. A power management circuit for adjusting the power consumption in a battery operated computer, the battery operated computer operating at a system clock frequency, comprising:

a counter-timer coupled to the battery operated computer for determining the rate of events;

a memory for storing a predetermined low event threshold and a predetermined high event threshold; and

a comparator coupled to the counter-timer and the memory, the comparator adapted to decrease the system clock frequency responsive to a rate of events being below the predetermined low event threshold and adapted to increase the system clock frequency responsive to the rate of events being above the predetermined high event threshold.

12. The power management circuit of claim **11**, wherein the events are selected from the group consisting of cache miss operations of the battery powered computer and

memory write operations of the battery powered computer, memory page miss operations of the battery powered computer and input/output write operations of the battery powered computer.

13. The power management circuit of claim 12, further comprising;

decode logic coupled to the counter-timer and the battery operated computer, the decode logic to select the type of event from which the rate of events is determined.

14. The power management circuit of claim 11, wherein the comparator signals the battery powered computer to decrease the system clock frequency when the rate of events is below the predetermined low event threshold and when the system clock frequency is at least a fast system clock frequency.

15. The power management circuit of claim 11, wherein the comparator signals the battery powered computer to increase the system clock frequency when the rate of events exceeds the predetermined high event threshold and when the system clock frequency is less than a slow system clock frequency.

16. A computer system capable of operation in different power consumption modes, the computer system operating in a normal power consumption mode when the computer system is subject to a normal level of activity and operating in a reduced power consumption mode when the computer system is subject to a reduced level of activity, comprising:

a processor operating at an adjustable system clock frequency;

a memory coupled to the processor; and

a power management circuit coupled to the processor, the power management circuit receiving the system clock frequency, comprising:

a clock frequency divider receiving the system clock frequency, the system clock frequency divider capable of outputting different output system clock frequencies, the output system clock frequency being a new system clock frequency, the clock frequency divider adapted to reduce the output system clock frequency responsive to a rate of processor events being below a predetermined low event threshold and adapted to increase the output system clock frequency responsive to the rate of processor events being above a predetermined high event threshold.

17. The computer system of claim 16, the power management circuit further comprising:

a comparator coupled to the clock frequency divider to determine when the rate of processor events is below a predetermined low event threshold and to determine when the rate of processor events exceeds a predetermined high event threshold.

18. The computer system of claim 16, further comprising:

a counter timer for determining the rate of processor events.

19. The computer system of claim 16, wherein the reduced output system clock frequency is output from the clock frequency divider when the rate of processor events is below the predetermined low event threshold and when the system clock frequency is at least a fast system clock frequency.

20. The computer system of claim 16, wherein the increased output system clock frequency is output from the clock frequency divider when the rate of processor events exceeds the predetermined high event threshold and when the system clock frequency is below a slow system clock frequency.

21. The computer system of claim 16, further comprising:

cache memory coupled to the processor; and

wherein the processor events comprise cache miss operations and memory write operations.

22. The computer system of claim 21, the power management circuit further comprising:

decode logic coupled to the processor, the decode logic to select the type of processor event from which the rate of processor events is determined.

23. The computer system of claim 16, further comprising:

cache memory coupled to the processor; and

input/output devices coupled to the processor

wherein the processor events are selected from the group consisting of cache miss operations and memory write operations, memory page miss operations and input/output write operations.

* * * * *

EXHIBIT C

US006438697B2

(12) **United States Patent**
Atkinson

(10) **Patent No.: US 6,438,697 B2**
(45) **Date of Patent: Aug. 20, 2002**

(54) **DEMAND-BASED PROCESSOR CLOCK FREQUENCY SWITCHING**

(75) Inventor: **Lee Warren Atkinson**, Houston, TX (US)

(73) Assignee: **Compaq Information Technologies Group, L.P.**, Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/818,981**

(22) Filed: **Mar. 27, 2001**

Related U.S. Application Data

(63) Continuation of application No. 09/480,101, filed on Jan. 10, 2000, now Pat. No. 6,233,691, which is a continuation of application No. 08/473,655, filed on Jun. 7, 1995, now Pat. No. 5,625,826, which is a continuation of application No. 08/166,609, filed on Dec. 13, 1993, now Pat. No. 6,029,249, which is a continuation of application No. 07/809,301, filed on Dec. 17, 1991, now abandoned.

(51) **Int. Cl.**[7] **G06F 1/32**
(52) **U.S. Cl.** **713/320**; 713/322; 713/501
(58) **Field of Search** 713/320–323, 713/501, 600, 300; 361/323, 683; 340/636; 307/60

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,085,449	A	4/1978	Walsh et al.
4,317,181	A	2/1982	Teza et al.
4,417,320	A	11/1983	Ei
4,531,826	A	7/1985	Stoughton et al.
4,670,837	A	6/1987	Sheets
4,698,748	A	10/1987	Juzswik et al.
4,819,164	A	4/1989	Branson
4,980,836	A	12/1990	Carter et al.
5,125,088	A	6/1992	Culley
5,142,684	A	8/1992	Perry et al.
5,163,143	A	11/1992	Culley et al.
5,167,024	A	11/1992	Smith et al.
5,218,704	A	6/1993	Watts, Jr. et al.
5,239,641	A	8/1993	Horst
5,487,181	A	1/1996	Dailey et al.
5,504,908	A	4/1996	Ikeda
5,655,127	A	8/1997	Rabe et al.
5,784,598	A *	7/1998	Griffith
6,029,249	A	2/2000	Atkinson
6,233,691	B1 *	5/2001	Atkinson

FOREIGN PATENT DOCUMENTS

EP	0451661	10/1991
JP	2151950	8/1990

OTHER PUBLICATIONS

Intel Corp., *386 SL Microprocessor Superset, System Design Guide*, Chapter 14, "System and Power Management," pp. 14–1 to 14–23, 1990.

Intel Corp., *396 SL Microprocessor Superset, Programmers Reference Manual*, Chapter 6 "Sytems and Power Management," pp. 6–1 to 6–50; 10–4 to 10–7, 1990.

Technique for Monitoring a Computer System's Activity for the Purpose of Power Management of a DOS–Compatible System, IBM Technical Disclosure Bulletin, vol. 33, No. 4, Sep. 1990, pp. 474–477.

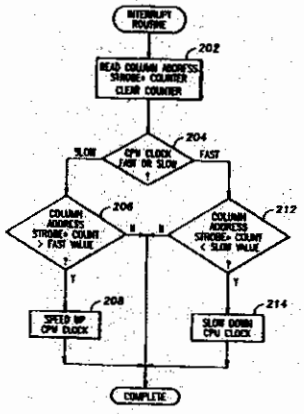
* cited by examiner

Primary Examiner—Gopal C. Ray
(74) *Attorney, Agent, or Firm*—Akin, Gump, Strauss, Hauer & Feld, LLP

(57) **ABSTRACT**

A battery powered computer system determines when the system is not in use by monitoring various events associated with the operation of the system. The system preferably monitors the number of cache read misses and write operations, i.e., the cache hit rate, and reduces the system clock frequency when the cache hit rate rises above a certain level. When the cache hit rate is above a certain level, then it can be assumed that the processor is executing a tight loop, such as when the processor is waiting for a key to be pressed and then the frequency can be reduced without affecting system performance. Alternatively, the apparatus monitors the occurrence of memory page misses, I/O write cycles or other events to determine the level of activity of the computer system.

26 Claims, 4 Drawing Sheets



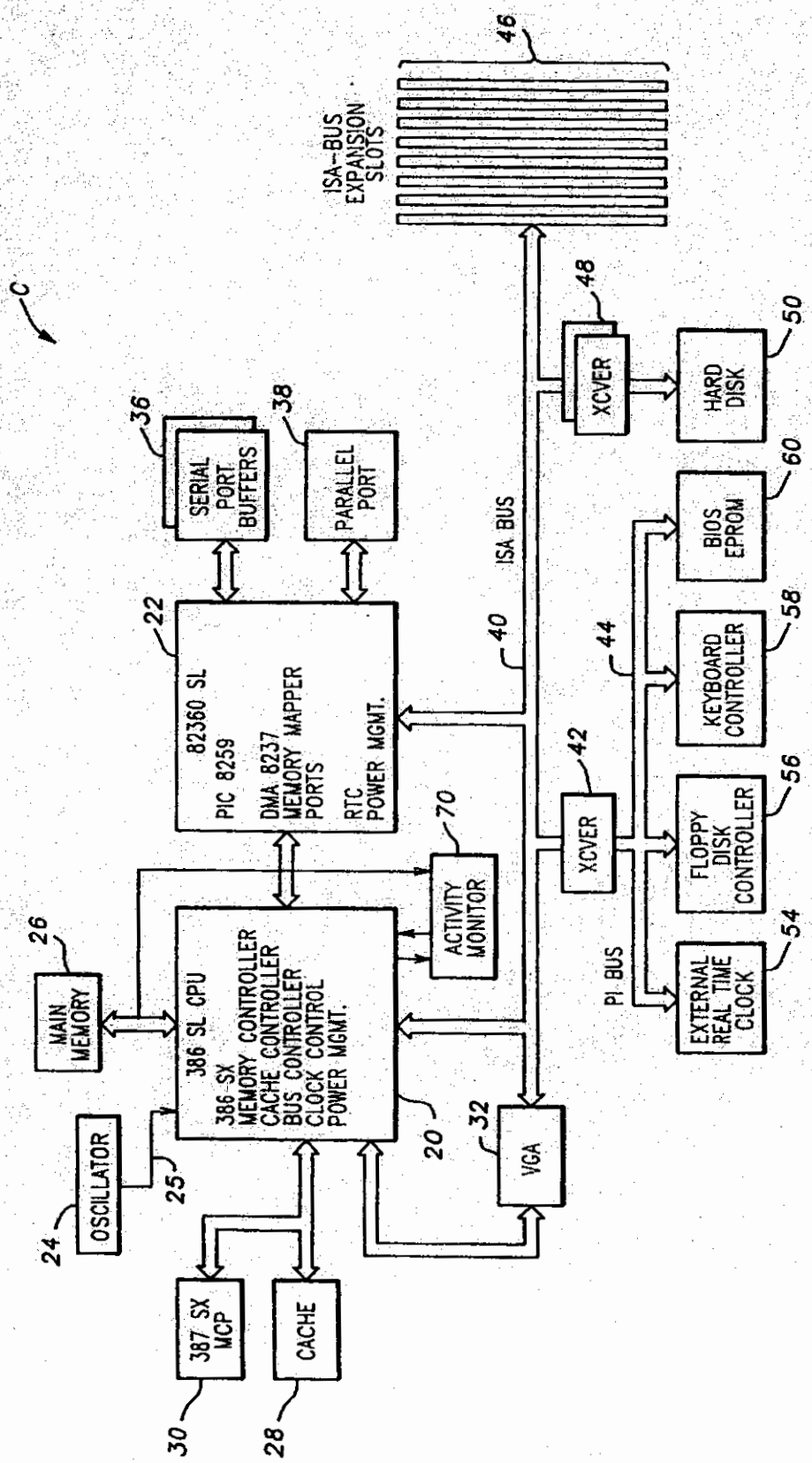

46
ISA-BUS EXPANSION SLOTS
C
48
XCVER
50
HARD DISK
36
SERIAL PORT BUFFERS
38
PARALLEL PORT
ISA BUS
60
BIOS EPROM
22
82360 SL
PIC 8259
DMA 8237
MEMORY MAPPER
PORTS
RTC
POWER MGMT.
40
44
58
KEYBOARD CONTROLLER
42
XCVER
56
FLOPPY DISK CONTROLLER
70
ACTIVITY MONITOR
26
MAIN MEMORY
386 SL CPU
386 SX
MEMORY CONTROLLER
CACHE CONTROLLER
BUS CONTROLLER
CLOCK CONTROL
POWER MGMT.
PI BUS
54
EXTERNAL REAL TIME CLOCK
20
32
VGA
24
OSCILLATOR
25
387 SX MCP
30
CACHE
28

FIG. 1

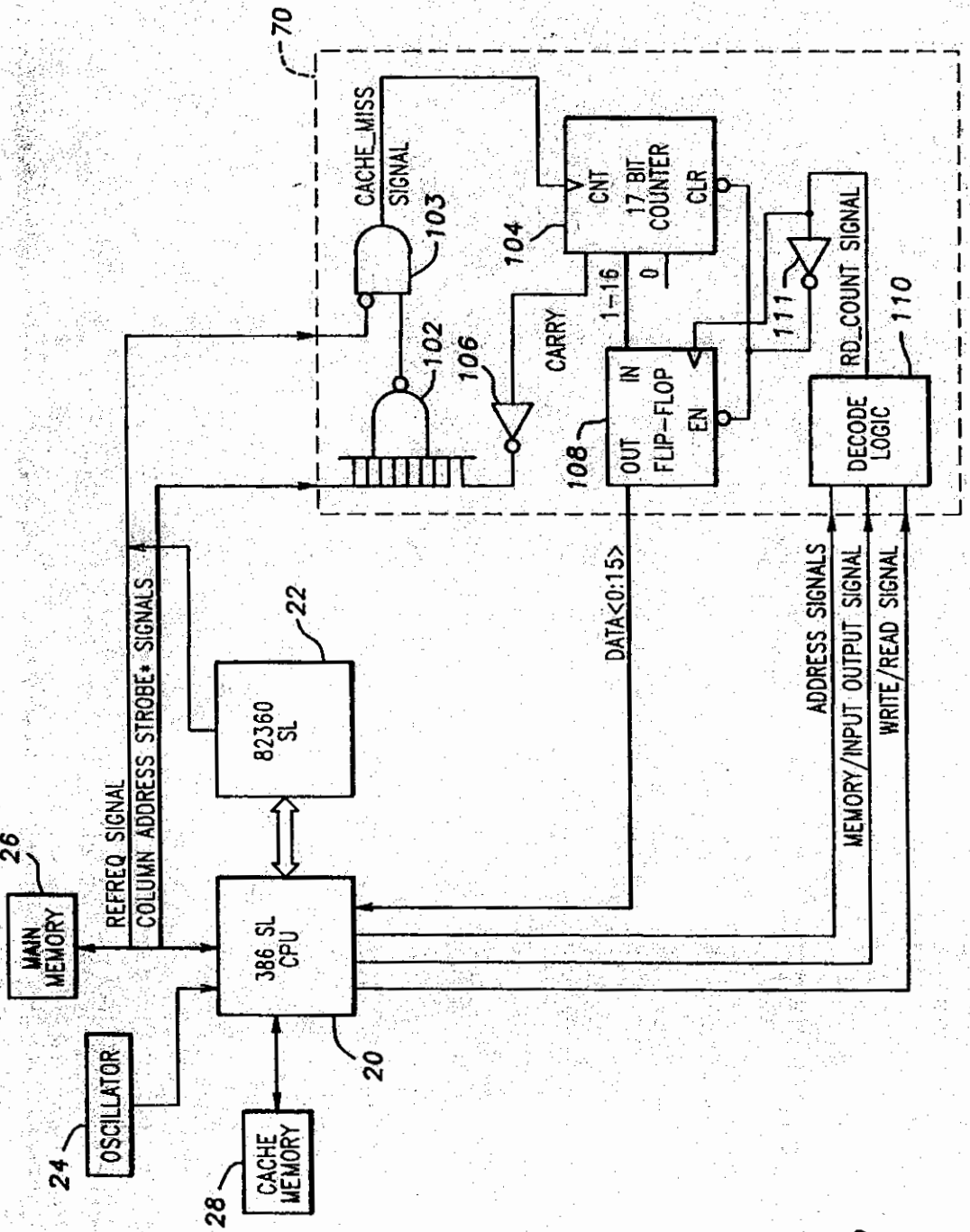

70
CACHE_MISS SIGNAL
103
104
CNT
17 BIT COUNTER
CLR
1-16
0
CARRY
102
106
111
RD_COUNT SIGNAL
110
108
IN
OUT
FLIP-FLOP
EN
DECODE LOGIC
DATA<0:15>
ADDRESS SIGNALS
MEMORY/INPUT OUTPUT SIGNAL
WRITE/READ SIGNAL
22
82360 SL
REFREQ SIGNAL
COLUMN ADDRESS STROBE* SIGNALS
26
MAIN MEMORY
386 SL CPU
20
OSCILLATOR
24
CACHE MEMORY
28

FIG. 2

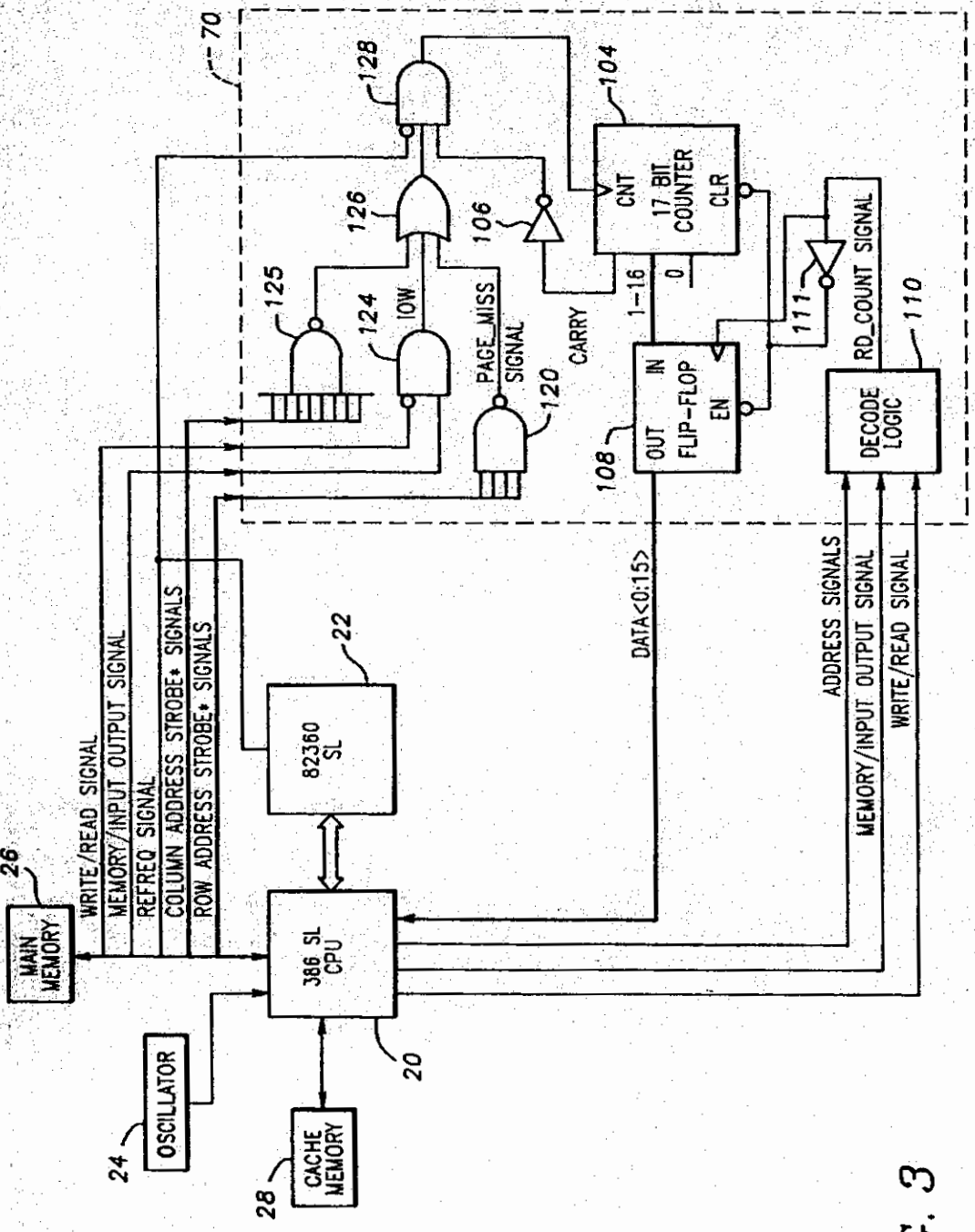

70
128
104
126
106
125
124
IOW
PAGE_MISS SIGNAL
120
CARRY
CNT
17 BIT COUNTER
CLR
1-16
0
111
RD_COUNT SIGNAL
110
IN
FLIP-FLOP
EN
OUT
108
DECODE LOGIC
DATA<0:15>
ADDRESS SIGNALS
MEMORY/INPUT OUTPUT SIGNAL
WRITE/READ SIGNAL
22
82360 SL
26
MAIN MEMORY
WRITE/READ SIGNAL
MEMORY/INPUT OUTPUT SIGNAL
REFREQ SIGNAL
COLUMN ADDRESS STROBE* SIGNALS
ROW ADDRESS STROBE* SIGNALS
386 SL CPU
20
OSCILLATOR
24
CACHE MEMORY
28

FIG. 3

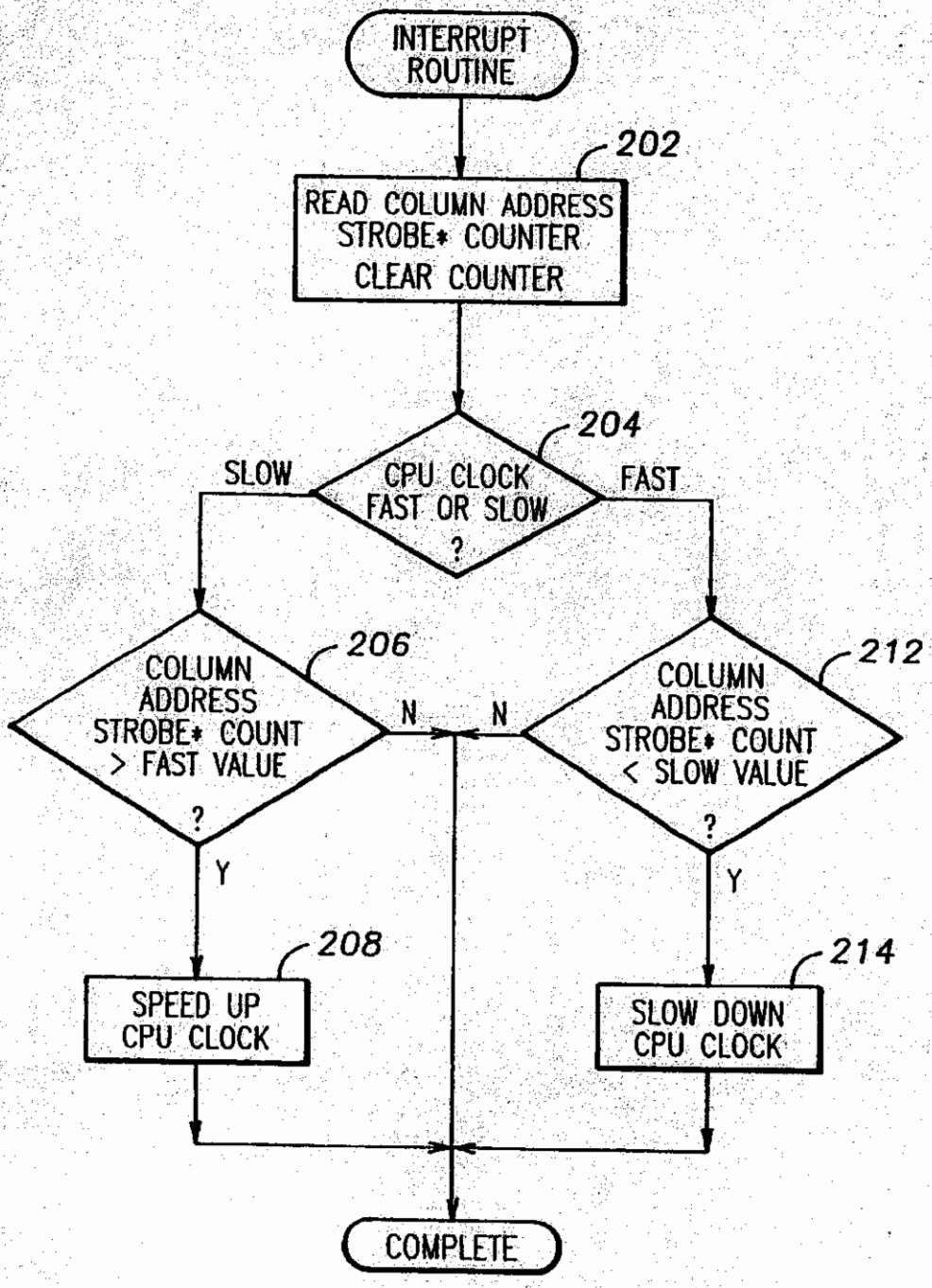

INTERRUPT ROUTINE
202
READ COLUMN ADDRESS STROBE* COUNTER
CLEAR COUNTER
204
CPU CLOCK FAST OR SLOW ?
SLOW
FAST
206
COLUMN ADDRESS STROBE* COUNT > FAST VALUE ?
N
N
212
COLUMN ADDRESS STROBE* COUNT < SLOW VALUE ?
Y
Y
208
SPEED UP CPU CLOCK
214
SLOW DOWN CPU CLOCK
COMPLETE

FIG. 4

DEMAND-BASED PROCESSOR CLOCK FREQUENCY SWITCHING

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of co-pending U.S. application Ser. No. 09/480,101, filed Jan. 10, 2000, now U.S. Pat. No. 6,233,691, which is a continuation U.S. application Ser. No. 08/166,609, filed Dec. 13, 1993, now U.S. Pat. No. 6,029,249, which is a continuation of U.S. application Ser. No. 08/473,655, filed Jun. 7, 1995, now U.S. Pat. No. 5,625,826, issued Apr. 29, 1997, which is a file wrapper continuation of U.S. application Ser. No. 07/809,301, filed Dec. 17, 1991, now abandoned, which are incorporated herein for reference.

STATEMENTS REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not applicable.

REFERENCE TO A MICROFICHE APPENDIX

Not applicable.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to battery powered computer systems, and more particularly, to circuits and methods for reducing the power consumption of the computer system.

2. Description of the Related Art

Portable computer systems are rapidly developing the capabilities of conventional desktop or floor mounted personal computer systems. Hard disk units are being integrated into portable computers because of the large amounts of information being processed and the large size of many application programs. A floppy disk unit is integrated in the vast majority of portable computers, even if a hard disk unit is installed, to allow loading of information and use of applications requiring key disks, and also to allow use of diagnostic programs. Modems have been integrated into portable computers for some time to allow communications and information transfer between the user and a remote location, for example, the home office. The displays in portable computer systems are becoming much more elaborate and readable. The pixel count on the standard liquid crystal displays (LCD's) utilized is increasing, as is the viewing angle. The use of backlighting allows use of LCD's in low light environments and improves the contrast ratio of the display. More complex circuitry is being installed in portable computers to support these improved peripheral devices and to support the increased speeds and capabilities of the microprocessors utilized in portable computer systems.

The various peripheral devices and high speed circuitry mentioned above consume large amounts of power when operating. This has resulted in problems in portable computer systems because these systems are generally desired to be used in locations where alternating current is not available. This has made it very difficult to provide all the possible functionality available and yet have an acceptable battery life when the portable computer system is battery powered. Using CMOS components helped reduce the power consumption of the circuitry, but even the use of CMOS components is insufficient at the clock speeds and performance levels of available circuitry. Therefore a dilemma arises whether to provide lesser functionality with longer battery life or greater functionality with lesser battery life or even no battery operation.

Various alternatives were tried to resolve the problem. For example, the International Business Machines (IBM) Corporation PC Convertible included a switch which the user could press to place the computer system in a standby mode. However, the PC convertible was relatively simple, with a low level of functionality as compared to what is currently available, and the requirement of a user action to reduce power consumption limited its use to circumstances where the user remembered to depress the switch. Blanking the display after a period of keyboard inactivity saved power as well as prolonged the life of the display and was widely utilized. A hard disk unit was developed which reduced the power used by the controlling electronics by utilizing only certain portions of the track for servo information and turning off the read channel circuitry until just before a servo burst was expected. Additionally, a programmable value could be provided to the hard disk unit so that after a given inactivity interval defined by this value, the hard disk unit was allowed to spin down and all but some interface circuitry was shut down. While these alternatives did provide some relief, they were not complete solutions to satisfactorily resolve the dilemma, and design tradeoffs still were forced to occur.

U.S. Pat. No. 4,980,836 to Carter et al. discloses an apparatus for reducing power consumption in computer systems. The apparatus monitors the address bus to determine when selected peripheral devices have not been accessed for a preset amount of time. When the preset amount of time has passed, the system powers itself down and disables the system clock, placing the system in a standby mode. The system clock could be stopped in this invention because the preferred embodiment of this invention used a static CMOS processor and chip set. If there was sufficient energy in the batteries, the system could be awakened by the user depressing a standby switch. Computer systems which do not use a static CMOS processor or chip set generally reduce the clock frequency when a preset amount of time of address bus inactivity has passed. Reducing the clock frequency during inactive periods reduces power consumption during this time. However, the frequency of peripheral device accesses is not a completely reliable indicator of inactivity of a computer system. Thus, in some instances the system clocking signal may be reduced in frequency or disabled during a period of high computer system activity. Therefore, a method is needed whereby other elements or events of the computer system can be monitored to more reliably determine the activity level of the computer system so that the system clock can properly be adjusted to reduce power consumption.

BRIEF SUMMARY OF THE INVENTION

A battery powered computer system according to the present invention determines when the system is not in use by monitoring various events associated with the operation of the computer system. In the preferred embodiment, the system monitors the number of cache read misses and write operations, i.e., the cache hit rate, and reduces the system clock frequency when the cache hit rate rises above a certain level. When the cache read hit rate is above a certain level, then it can be assumed that the processor is executing a tight loop, such as when the processor is waiting for a key to be pressed. In an alternate embodiment of the invention, the system monitors other events in addition to, or instead of, the cache read hit rate, such as the occurrence of page hits or input/output (I/O) write cycles, to determine the level of activity of the computer system.

The system according to the preferred embodiment includes a frequency switching circuit, an event counter, and a periodic timer. The event counter is preferably used to measure the incidence of cache read misses and write operations and may also optionally be used to count the number of page misses and memory or I/O writes as desired. The event counter includes an overflow or carry line which prevents any further incrementing of the counter once the maximum number of counts is reached to prevent the counter from overflowing. The periodic timer instructs the CPU via a system interrupt to periodically monitor and compare the contents of the event counter. Every event increments the counter and, the more events, the more processor activity that is presumed. When the periodic timer issues a system interrupt, the CPU reads the contents of the counter and compares the event activity with a predetermined value. If the number of events is higher than the predetermined value, then the processor switches the operating frequency of the system to a high frequency if the system is not already operating at this high frequency. A lower event count causes the frequency switching circuit to switch to a lower frequency to conserve power if the system is not already operating at this low frequency.

The invention allows the battery powered operating period of a computer system to be greatly extended without requiring any input from the user and without any noticeable loss in processing power. This allows a battery powered computer system to have advanced capabilities and functionality while still having a satisfactory battery operating interval.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

A better understanding of the invention can be obtained when the following detailed description of the preferred embodiment is considered in conjunction with the following drawing in which:

FIG. **1** is a schematic block diagram of a computer system incorporating the present invention;

FIG. **2** is a more detailed schematic diagram of a portion of the computer system of FIG. **1**;

FIG. **3** is a schematic diagram of a portion of the computer system of FIG. **1** according to an alternate embodiment of the invention; and

FIG. **4** is a flowchart diagram of a sequence for controlling the operation of the computer system of FIG. **1** according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to FIG. **1**, a computer system C according to the preferred embodiment of the invention is shown. The computer system C is preferably based on the 386 SL chip set produced by Intel Corporation (Intel). The 386 SL chip set comprises two chips, a 386 SL CPU chip (CPU chip) **20** and a 82360 SL chip **22**. The CPU chip **20** includes an 80386 SX microprocessor, a memory controller, a cache controller, a bus controller, clock control circuitry, and power management circuitry. The 82360 SL chip **22** includes a programmable interrupt controller (PIC), direct memory access (DMA) controller, a memory mapper, various ports, a real time clock (RTC) and power management circuitry. For more information on the 386 SL chip set, please see the Intel 386 SL Microprocessor Superset System Design Guide 1990 edition; the 386 SL Microprocessor Superset Programmers

Reference Manual, 1990 edition; and the 386 SL Microprocessor Superset Data Book; all published by Intel.

An oscillator **24** is connected to the CPU chip **20**. The oscillator preferably operates at 20 Megahertz (MHz) and provides a 20 MHz clocking signal **25** to the CPU chip **20**. Main memory **26**, cache memory **28** and a math coprocessor (MCP) **30** are also preferably coupled to the CPU chip **20**. The math coprocessor **30** is preferably a 387 SX coprocessor produced by Intel. The cache memory is preferably operated as a write-through cache memory. A video graphics array (VGA) controller **32** is coupled to the CPU chip **20**. Serial port buffers **36** and a parallel port **38** are coupled to the 82360 SL chip **22**.

The CPU chip **20** and the 82360 SL chip **22** are each connected to an I/O bus **40** based on the industry standard architecture (ISA). However, other bus architectures are also contemplated. The ISA bus **40** is connected through a transceiver **42** to a peripheral interface (PI) bus **44**. The VGA chip **32** is connected to the ISA bus **40**. The ISA bus **40** may include a plurality of ISA bus expansion slots **46** if the present invention is used in a desktop computer system. The slots **46** are generally omitted in portable computers. The ISA bus **40** is connected through a transceiver **48** to a hard disk **50**. Various logic is coupled to the PI bus **44**, including an external real time clock **54**, a floppy disk controller **56**, a keyboard controller **58** and BIOS EPROM **60**. The floppy disk controller **56** and keyboard controller **58** control operation of a floppy disk unit and keyboard, respectively (both not shown). The BIOS EPROM holds the basis input/output (I/O) system software as well as system-specific initialization and configuration software. This is an exemplary computer system and other designs and architectures could be utilized.

The CPU chip **20** includes an active high input referred to as the TURBO input (not shown). When the TURBO input receives a logic high signal, the CPU chip **20** enters "turbo mode," and the microprocessor executes at a clock speed defined by a bit field in a register in the CPU chip **20** referred to as the CPUWRMODE register (not shown). When the TURBO input receives a logic low signal, the CPU chip **20** enters "de-turbo mode" and executes at a reduced clock speed of ½ or ¼ as defined by a bit in a register. The CPUWRMODE register operates in conjunction with the power management circuitry inside the CPU chip **20** to control the frequency of the clocking signal **25** provided to the microprocessor. In the preferred embodiment, the turbo input receives a logic high value, and thus the CPUWRMODE register determines the frequency of the clocking signal **25** provided to the microprocessor.

Bits **5** and **4** of the CPUWRMODE register determine the speed with which the microprocessor clock operates. When bits **5** and **4** of the CPUWRMODE register are each set to 0, then the clock speed is the speed of the signal received by the oscillator **24**, preferably 20 MHz. When bits **5** and **4** of the CPUWRMODE register are set to 0 and 1 respectively, the clock speed provided to the microprocessor is one half of the frequency of the signal received from the oscillator **24**, i.e., 10 MHz. When bits **5** and **4** of the CPUWRMODE register are 1 and 0 respectively, the clock speed provided to the microprocessor is one fourth of the frequency of the signal received from the oscillator **24**, i.e., 5 MHz. When bits **5** and **4** of the CPUWRMODE register each have a logic 1 value, then the clock speed provided to the CPU is one eighth the frequency of the signal received from the oscillator **24**, i.e. 2.5 MHz.

The computer system C also includes activity monitor logic **70** according to the present invention which is coupled

to the CPU chip **20**. The activity monitor logic **70** receives signals from the CPU chip **20** as well as signals provided from the CPU chip **20** to the main memory **26**. The activity monitor logic **70** monitors events associated with the microprocessor to determine the activity level of the system. In the preferred embodiment, the activity monitor logic **70** monitors the number of cache read misses and write operations during preset periods of time, and the CPU chip **20** reduces the system clock frequency when the cache hit rate rises above a certain level. When the cache read hit rate is above a certain level, then it can be assumed that the processor is executing a tight loop, such as when the processor is waiting for a key to be pressed. In this instance, the system clock frequency can be reduced to reduce power consumption without affecting system performance. In an alternate embodiment of the invention, the activity monitor logic **70** monitors other events such as page misses or memory and I/O write operations, among others. It is contemplated that the various types of events may be counted either individually or in combination.

Referring now to FIG. **2**, the activity monitor logic **70** according to the preferred embodiment of the invention is shown. In the description that follows a signal name followed by an asterisk indicates that the signal is asserted when it has a logic low level. The activity monitor logic **70** receives column address strobe (CAS*) signals provided from the CPU chip **20** to the main memory **26**. The CAS* signals are essentially memory select signals that are used in paged memory systems. When a cache read hit occurs, the requested data can be provided directly from the cache memory **28** to the microprocessor, and no CAS* signals are asserted by the CPU chip **20**. Also, since the cache memory is operated as a write-through cache, each write operation, whether it be a write hit or write miss, requires a memory cycle to the main memory **26**. Therefore, the CAS* signals are asserted on all write operations in the preferred embodiment. Thus, the CAS* signals are used to determine when cache read miss operations and write operations occur, i.e., when main memory cycles occur.

In the preferred embodiment, the CPU chip **20** generates eight CAS* signals. The CAS* signals are provided to eight inputs of a nine input NAND gate **102**, in effect ORing these signals together. The output of the NAND gate **102** is provided to an input of a two input AND gate **103**. A signal referred to as REFREQ output from the 82360 SL chip **22** is preferably provided to the other input of the AND gate **103**, which is an inverted input. The REFREQ signal indicates, when asserted high, that a memory refresh operation is occurring. The inverted REFREQ signal is ANDed with the output of the NAND gate **102** to prevent memory refresh operations from being counted as cache read miss or write operations.

The output of the AND gate **103** is a signal referred to as CACHE_MISS, which is provided to the clock input of a 17 bit counter **104** referred to as the CAS* counter. The CACHE_MISS signal is therefore asserted when a main memory cycle occurs, which is whenever a cache read miss or a write operation occurs. The CAS* counter **104** is incremented each time the CACHE_MISS signal is asserted. The CAS* counter **104** outputs a signal referred to as CARRY which is provided through an inverter **106** to the seventh input of the NAND gate **102**. The CARRY signal is asserted when the maximum number of counts is reached by the counter **104**, which is 2^{17}. When the CARRY signal is asserted, the CACHE_MISS signal is prevented from changing state and thus further counting by the counter **104** is disabled. The CAS* counter **104** provides data signals **1–16** to the inputs of a 16 bit flip-flop **108**. Data signal **0** output from the CAS* counter is preferably left unconnected. The outputs of the flip-flop **108** are coupled to data bus signals referred to as DATA<0:15>, which are coupled to the CPU chip **20**.

Address signals, a write/read (W/R) signal and a memory/input output (M/IO) signal are output from the CPU chip **20** and provided to decode logic **110** in the activity monitor logic **70**. The decode logic **110** outputs a signal referred to as RD_COUNT which is provided to a clock input of the flip-flop **108**. The RD_COUNT signal is also provided through an inverter **111** to an inverted clear input of the counter **104** and to an inverted enable input of the flip-flop **108**. Therefore, when the CPU chip **20** desires to read the counter **104**, it outputs the appropriate address, the W/R signal and the M/IO signal to the decode logic **110**, which then asserts the RD_COUNT signal to the counter **104** and the flip-flop **108**. The asserted RD_COUNT signal enables the flip-flop **108** to latch in data from the counter **104** and transmit the data to the CPU chip **20** and simultaneously operates to clear the contents of the counter **104**.

In an alternate embodiment, memory page misses and/or I/O write operations may also be counted. Memory page miss operations with the preferred microprocessor are determined by determining if any of the 4 row address strobe or RAS* signals are asserted low during non-refresh cycles. The asserted RAS* signals are indicative of page miss operations. As shown in FIG. **3**, the RAS* signals are provided from the CPU chip **20** to the inputs of a four input NAND gate **120**. The output of the NAND gate **120** is the page miss indication, which is referred to as the PAGE_MISS signal. The I/O write indication is accomplished in the following manner. The W/R signal is connected to an input of a two input AND gate **124**. The other input of the AND gate **124** is inverted and receives the M/IO signal. The output of the AND gate **124** generates a signal referred to as IOW which indicates, when asserted high, that an I/O write operation is occurring.

The PAGE_MISS and IOW signals are connected to inputs of a three input OR gate **126**. The eight CAS* signals are connected to the input of an eight input NAND gate **125** whose output is connected to an input of the OR gate **126**. The output of the OR gate **126** is connected to an input of a three input AND gate **128**. A second input of the AND gate **128** receives the inverted CARRY signal. The third inverted input of the AND gate **128** receives the REFREQ signal. The output of the AND gate **128** is provided to the clock input of the counter **104**. Therefore, during memory refresh operations or when the CARRY signal is asserted, the output of the AND gate **128** is negated low, and thus the counter **104** does not count at these times.

In this embodiment, cache read miss and write operations, page miss operations, and I/O write operations are all monitored as events. It is noted that any one of these events can be monitored either individually or in combination. In yet another alternate embodiment, each signal can be provided to separate counters arranged like the counter **104**, and the outputs of the separate counters can be used to determine system activity.

The 82360 SL chip **22** includes an interrupt timer (not shown) which is used to generate time of day interrupts to the microprocessor approximately 18.2 times per second. The interrupt routine invoked by the timer is also used according to the preferred embodiment of the invention to periodically adjust the frequency of the clocking signal **25** received by the microprocessor, as is explained below.

Referring now to FIG. 3, a portion of the software routine that is executed when the time of day interrupt routine is invoked is shown. It is understood that the time of day interrupt routine may perform other operations than those shown. When the time of day interrupt signal is asserted, the CPU chip **20** reads the CAS* counter **104** in step **202**. The counter **104** is also cleared in step **202**. In an alternative embodiment, the separate counters providing the CAS* count, the memory page miss count, and the I/O write count are read in step **202**. In step **204**, the microprocessor reads bits **5** and **4** of the CPUWRMODE register to determine if the clock signal provided to the CPU chip **20** is operating at a fast or slow frequency. In the preferred embodiment, the microprocessor only writes either values 0, 0 or 1,0 to bits **5** and **4**, respectively, of the CPUWRMODE register, thus providing the clocking signal **25** at either full frequency or one quarter frequency to the microprocessor. The CPUWRMODE register is programmed depending on the cache read hit rate as determined by the CAS* counter **104** in relation to certain comparison values. Thus, in this embodiment, the power management logic toggles between a full clock speed or fast speed, preferably 20 MHz, and a one quarter frequency clock speed or slow speed, preferably 5 MHz. However, it is noted that all four clock speed frequencies may be utilized.

If the clocking signal **25** provided to the CPU is determined to be operating at the slow speed in step **204**, then in step **206** the microprocessor determines if the CAS* count value received from the CAS* counter **104** is greater than a fast comparison value stored inside the CPU chip **20**. Alternatively, a function based on the three separate count values could be evaluated. If the CAS* count value exceeds the fast comparison value in step **206**, then in step **208** the power management logic in the CPU chip **20** speeds up the clock signal **25** provided to the microprocessor to the full clock speed, and the routine completes. If the CAS* count value is not greater than the fast comparison value in step **206**, then the interrupt routine completes.

If the clocking signal **25** provided to the microprocessor is determined to be operating at the fast speed in step **204**, then control proceeds to step **212** where the microprocessor determines whether the CAS* count value is less than a slow comparison value stored in the CPU chip **20**. If the CAS* count value is less than the slow comparison value in step **212**, then the power management logic in the CPU chip **20** slows down the clocking signal **25** provided to the microprocessor in step **214** to one quarter speed, and the routine completes. If the CAS* count value is not less than the slow comparison value in step **212**, then the routine completes.

The fast and slow comparison values are preferably programmable inside the CPU chip **20**. In the preferred embodiment, the fast and slow comparison values are the same value. In an alternate embodiment, the slow comparison value is proportionally less that the fast value, i.e., one fourth less to reflect a true hit/miss ratio. In addition, since the comparison values can be user-adjusted inside the CPU chip **20**, the values could be derived from characterization of common user applications.

As described above, the preferred embodiment utilizes the CPU chip **20** in "turbo mode" and uses software to perform the activity comparisons and speed changes. In an alternative embodiment, additional registers can be used to contain the comparison values and an additional timer can be used to define the comparison interval. Hardware comparators are configured to perform the comparison logic described above at the appropriate time and change the state of the TURBO input based on the comparison to speed up or slow down the CPU chip **20** as appropriate.

Thus, the computer system according to the present invention monitors various power consumption related events, indicates certain changes to the user and enters an inactivity state upon an appropriate period of time after monitored system devices have been used.

The foregoing disclosure and description of the invention are illustrative and explanatory thereof, and various changes in the components, circuit elements, or flowcharts may be made without departing from the spirit of the invention.

I claim:

1. A computer system, comprising:

a processor;

a means for monitoring activity of the processor; and

a means for switching the processor from a first processor clock frequency to a second processor clock frequency based on the activity of the processor.

2. The computer system of claim **1**, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to the second processor clock frequency when the processor activity is below a predetermined processor activity threshold.

3. The computer system of claim **1**, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to the second processor clock frequency when the processor activity is above a predetermined processor activity threshold.

4. The computer system of claim **1**, wherein the second processor clock frequency is one of a plurality of selectable processor clock frequencies.

5. The computer system of claim **1**, wherein the second processor clock frequency is less than the first processor clock frequency.

6. The computer system of claim **1**, wherein the second processor clock frequency is greater than the first processor clock frequency.

7. A method of power management, comprising the steps of:

monitoring activity of a processor of a computer system; and

switching the processor from a first processor clock frequency to a second processor clock frequency based on the activity of the processor.

8. The method of claim **7**, the switching step comprising the step of:

reducing the processor from the first processor clock frequency to the second processor clock frequency.

9. The method of claim **7**, the switching step comprising the step of:

increasing the processor from the first processor clock frequency to the second processor clock frequency.

10. The method of claim **7**, wherein the computer system comprises a portable computer system.

11. A power management system, comprising:

a means for monitoring activity of a processor of a computer system; and

a means for switching the processor from a first processor clock frequency to a second processor clock frequency based on the activity of the processor.

12. The system of claim **11**, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to the second processor clock frequency when the processor activity is below a predetermined processor activity threshold.

13. The system of claim 11, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to the second processor clock frequency when the processor activity is above a predetermined processor activity threshold.

14. The system of claim 11, wherein the second processor clock frequency is one of a plurality of selectable processor clock frequencies.

15. A computer system, comprising:

a processor;

a means for monitoring a processor signal associated with the processor wherein the processor signal represents a count; and

a means for switching the processor from a first processor clock frequency to a second processor clock frequency based on the processor signal.

16. The computer system of claim 15, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to the second processor clock frequency when the count is greater than a predetermined value.

17. The computer system of claim 15, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to the second processor clock frequency when the count is less than a predetermined value.

18. The computer system of claim 15, wherein the second processor clock frequency is one of a plurality of selectable processor clock frequencies.

19. A method of power management, comprising the steps of:

monitoring a processor signal associated with a processor of a computer system wherein the processor signal represents a count; and

switching the processor from a first processor clock frequency to a second processor clock frequency based on the processor signal.

20. The method of claim 19, the switching step comprising the step of:

reducing the processor from the first processor clock frequency to the second processor clock frequency.

21. The method of claim 19, the switching step comprising the step of:

increasing the processor from the first processor clock frequency to the second processor clock frequency.

22. The method of claim 19, wherein the computer system comprises a portable computer system.

23. A power management system, comprising:

a means for monitoring a processor signal associated with a processor of a computer system wherein the processor signal represents a count; and

a means for switching the processor from a first processor clock frequency to a second processor clock frequency based on the processor signal.

24. The system of claim 23, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to a second processor clock frequency when the count is less than a predetermined value.

25. The system of claim 23, the means for switching comprising:

a means for switching the processor from the first processor clock frequency to the second processor clock frequency when the count is greater than a predetermined value.

26. The system of claim 23, wherein the second processor clock frequency is one of a plurality of selectable processor clock frequencies.

* * * * *